Introduction to linear logic

Emmanuel Beffara

Introduction to linear logic

Emmanuel Beffara

August 29, 2013

Abstract. This manuscript is the lecture notes for the course of the same title I gave at the *Summer school on linear logic and geometry of interaction* that took place in Torino in August 2013. The aim of this course is to give a broad introduction to linear logic, covering enough ground to present many of the ideas and techniques of the field, while staying at a (hopefully) accessible level for beginners. For this reason, most technical development is carried out in the simple multiplicative fragment, with only hints at generalizations. As prerequisites, some knowledge of classical sequent calculus and some knowledge of the λ-calculus is useful.

Reference material on the topics discussed here include

Girard, Lafont, and Taylor, *Proofs and types* is a book evolved from lecture notes of a graduate course of the same title. It is a good reference for an introduction to the proof-program correspondence, although it does not cover all topics of the matter.

Girard, "Linear Logic" is the historical paper introducing linear logic. It is an unavoidable reference, although not the best way to discover the topic nowadays, since many aspects have been better understood since then.

Girard, "Linear Logic: Its Syntax and Semantics" is an updated and more accessible presentation, written about ten years later.

1 The proof-program correspondence

Logic is the study of discourse and reasoning. Mathematical logic is the study of mathematical discourse and reasoning. The need for mathematical logic historically arose at the end of the nineteenth century with the search for foundations of mathematics, at a time when abstract mathematics reached an unprecedented level of complexity and stumbled upon paradoxes. Proof theory is the sub-field of mathematical logic concerned with the nature and meaning of mathematical statements and proofs. Gödel's incompleteness theorem and Gentzen's cut-elimination method are the first major results that contributed in its definition in the 1930s. The following years saw the progressive apparition of the central role of computation in logic, up to the formulation of the Curry-Howard isomorphism in the 1960, as briefly presented below. This correspondence is now a central notion both in proof theory and in theoretical computer science as it addresses the central question of both fields, which is *consistency*:

- A logical system is consistent if it is not degenerate, i.e. if it does not prove everything. Then one can search for a *meaning* of proofs, and the next level of consistency is if there are statements for which there are different proofs, hence different ways of proving things.
- Consistency in a formal language for computation refers to the ability, by structural considerations like typing, to make sure that a program behaves well (no deadlocks, no infinite loop). This in turn implies the definition of the *meaning* of programs.

This search for meaning is known as *semantics* and is the meeting point between logic and computation. Linear logic is one of the outcomes of the study of semantics and the interaction between logic and computer science. See Girard, Lafont, and Taylor, *Proofs and types*, as a more detailed introduction to this topic.

1.1 The Curry-Howard isomorphism

The clearest formulation of the proof-program correspondence, also known as the Curry-Howard isomorphism, is obtained through the simply typed λ-calculus. We present it here briefly, in its extension with conjunction types.

In the following, we will assume that one is given, once and for all, a set of propositional variables (ranged over by Greek letters $\alpha, \beta, ...$) and a set of term variables (ranged over by Latin letters $x, y, z...$).

1 **Definition**. Formulas of implicative-conjunctive propositional logic are defined by the following grammar:

$$\begin{array}{rll} A, B := & \alpha & \text{propositional variables} \\ & A \Rightarrow B & \text{implication} \\ & A \wedge B & \text{conjunction} \end{array}$$

Terms of the simply-typed λ-calculus with pairs are defined by the following grammar:

$$\begin{array}{rll} t, u := & x & \text{variable} \\ & \lambda x^A.t & \text{abstraction, i.e. function} \\ & (t)u & \text{application} \\ & \langle t, u \rangle & \text{pairing} \\ & \pi_i t & \text{projection, with } i = 1 \text{ or } i = 2 \end{array}$$

where x in $\lambda x^A.t$ is a bound variable. Terms are considered up to injective renaming of bound variables, assuming all bound variables in a given term are distinct from all free variables.

A typing context is a sequence $x_1 : A_1, ..., x_n : A_n$ where the variables $x_1, ..., x_n$ are all distinct. A typing judgment consists in a typing context Γ, a λ-term t and a formula A and is written $\Gamma \vdash t : A$. A term has type A in a context Γ if the judgment $\Gamma \vdash t : A$ can be proved using the rules of table 1.

$$\frac{}{\Gamma, x : A \vdash x : A}\ \text{ax} \qquad \frac{\Gamma, x : A \vdash t : B}{\Gamma \vdash \lambda x^A.t : A \Rightarrow B}\ \Rightarrow\text{I} \qquad \frac{\Gamma \vdash t : A \Rightarrow B \quad \Gamma \vdash u : A}{\Gamma \vdash (t)u : B}\ \Rightarrow\text{E}$$

$$\frac{\Gamma \vdash t : A \quad \Gamma \vdash u : B}{\Gamma \vdash \langle t, u\rangle : A \wedge B}\ \wedge\text{I} \qquad \frac{\Gamma \vdash t : A \wedge B}{\Gamma \vdash \pi_1 t : A}\ \wedge\text{E1} \qquad \frac{\Gamma \vdash t : A \wedge B}{\Gamma \vdash \pi_2 t : B}\ \wedge\text{E2}$$

Table 1: Typing rules for the simply typed λ-calculus

In the rules of table 1, if we forget everything that is on the left of colons, i.e. all λ-terms and variables, we get a deduction system for implicative-conjunctive propositional logic. This system is known as intuitionistic *natural deduction* and called NJ (actually NJ refers to the whole natural deduction system for intuitionistic logic, of which this is a simple subsystem). If we read $A_1, ..., A_n \vdash B$ as "under hypotheses $A_1, ..., A_n$, I can prove B", then this system is correct.

Now if we look only at λ-terms, we get a minimal functional language. An abstraction $\lambda x^A.t$ intuitively represents the function which, to an argument x of type A, associates the value of t. Computation in this language is formalized by a reduction relation $\rightsquigarrow$ defined by two simple rules which apply anywhere in a term:

$$(\lambda x.t)u \rightsquigarrow t[u/x] \qquad\qquad \pi_i\langle t_1, t_2\rangle \rightsquigarrow t_i \quad \text{for } i = 1 \text{ or } i = 2$$

where $t[u/x]$ is the operation of replacing every occurrence of the variable x in t by the term u.

The fact that computation is correct, i.e. respects types and produces no infinite loops, is formulated by the following theorems, of which we will provide no proof in this introduction.

2 **Theorem (Subject reduction).** *If $\Gamma \vdash t : A$ holds and $t \rightsquigarrow u$ then $\Gamma \vdash u : A$ holds.*

3 **Theorem (Confluence).** *For any pair of reductions $t \rightsquigarrow^* u$ and $t \rightsquigarrow^* v$ of a term t, there exists a term w and a pair of reductions $u \rightsquigarrow^* w$ and $v \rightsquigarrow^* w$.*

4 **Theorem (Strong normalization).** *A typable term has no infinite sequence of reductions.*

Theorems 3 and 4 together imply that any λ-term has a unique irreducible reduct and that any sequence of reduction steps eventually reach it.

In an irreducible typed term, by definition of reduction, there is never an introduction rule (for either $\Rightarrow$ or $\wedge$) followed by an elimination rule for the same connective. Proofs with this property will be called *normal* and they have the following crucial property:

5 **Theorem (Subformula property).** *In a normal proof, any formula occurring in a sequent at any point in the proof is a subformula of one of the formulas in the conclusion.*

The reduction operation from the λ-calculus, interpreted on proofs, is a normalization procedure that computes *the* normal form of an arbitrary proof. The effect of this, as illustrated by the subformula property, is that a proof is made *explicit* in the process, in other words it transforms an arbitrary proof into a direct proof, without lemmas. Indeed, a lemma for a theorem is a statement that is usually not a part of the theorem's statement but is proved as an intermediate step for proving the theorem.

1.2 Denotational semantics

Reduction in the λ-calculus can be seen as a computation method in a world of "pure" functions, indeed the λ-calculus was initially introduced by Church in order to define a pure theory of functions, in a way analogous to set theory which describes a world made entirely of sets. It is natural to ask whether this theory has a model where the terms are actually interpreted as functions and *denotational semantics* is precisely the study of such interpretations. By the Curry-Howard isomorphism, this will also interpret

proofs as functions, for instance a proof of $A \Rightarrow B$ will actually be a function from proofs of A to proofs of B. The key ingredients to make this concrete are the following:

- types/formulas are interpreted by spaces (often as particular sets with additional structure, more generally by objects in a suitable category),
- terms/proofs are interpreted as morphisms between such spaces,
- reduction preserves the interpretation of terms.

The last point is the most important one: if syntactic reduction actually computes something, then reducing means getting closer to the final syntactic form of the computed object, but the object itself does not change. We will describe here a particular model of the simply-typed λ-calculus known as *coherence spaces*.[1]

6 *Example.* This simplest and most naive model is obtained by interpreting formulas as plain sets and terms as (set-theoretic) functions. Propositional variables are arbitrary sets, $A \Rightarrow B$ is interpreted by the set of functions from A to B. This models perfectly well but it does not have much interest because it is not informative about what proofs and terms mean. That is because there are way too many morphisms, most of which exhibit behaviours that one could not be defined using terms. A symptom is that this model fails when considering second order quantification (i.e. quantification over types, also known as polymorphism), because of cardinality problems.

7 **Definition**. A *coherence space* A is a set $|A|$ (its web) and a symmetric and reflexive binary relation $\frown_A$ (the coherence). A *clique* in A is a subset of $|A|$ of points pairwise related by $\frown_A$. The set of cliques of A is written $\mathcal{C}\ell(A)$.

The idea is that $|A|$ is a set of possible bits of information about an object of type A and the coherence relation $\frown_A$ indicates which bits can possibly describe aspects of the same object. A clique is a set of mutually coherent bits, hence a (possibly partial) description for some object.

8 *Example.* We could represent a type of words over some alphabet using a web $|A|$ made of pairs (i, x) that mean "at position i there is a letter x" and pairs $(n, \$)$ that mean "at position n is the end-of-string symbol". Coherence will be defined as $(i, x) \frown (j, y)$ if either $i = j$ then $x = y$ or $i < j$ and $x \neq \$$ or $i > j$ and $y \neq \$$.

9 **Definition**. Given coherence spaces A and B, a stable function from A to B is a function f from $\mathcal{C}\ell(A)$ to $\mathcal{C}\ell(B)$ that is

- continuous: if $(a_i)_{i\in I}$ is a directed family in $\mathcal{C}\ell(A)$, then $f(\bigcup_{i\in I} a_i) = \bigcup_{i\in I} f(a_i)$;
- stable: for all $a, a' \in \mathcal{C}\ell(A)$ such that $a \cup a'$ is a clique, $f(a \cap a') = f(a) \cap f(a')$.

These conditions imply monotonicity: the more information we have on the input, the more information we have on the output. Continuity means that the value for an infinite input can be deduced from the values for its finite approximations. Stability is more subtle, it means that if a bit is available in the output, then there is a minimal set of bits in the input needed to get it.

10 **Definition**. The *trace* of a stable function f from A to B is the set

$$Tr\big(f\big) := \Big\{(a, \beta) \;\Big|\; \beta \in f(a) \wedge \forall a' \subsetneq a, \beta \notin f(a')\Big\}.$$

where a and a' are cliques in the coherence space A and β is a point in the coherence space B.

[1] This model was first introduced under the name of "binary qualitative domains" as a semantics for System F, i.e. the simply typed λ-calculus extend with quantification over types, in Girard, "The system F of variable types, fifteen years later".

In each pair (a, β) in $Tr\left(f\right)$, a is the minimal input (a clique in A) needed to produce β (a point in B). It can be checked that this trace fully defines the stable function, hence there is a bijection between stable functions and traces.

Furthermore, it appears that traces are cliques in a suitable coherence space whose web is made of all pairs of a finite clique in A and a point in B. Coherence between two pairs simply expresses that they may be part of the same stable function. This coherence space is how we interpret the implication $A \Rightarrow B$. An interpretation of simply typed λ-terms as cliques can be deduced from these considerations.

11 **Definition**. A stable function f is *linear* if for all $(a, \beta) \in Tr\left(f\right)$, a is a singleton.

In other words, for producing one bit of information in output, the function needs one bit of information in input. This notion corresponds, in a way that can be made very precise, to the fact the f uses its argument *exactly once* in order to produce each bit of information in output. This is a very computational property, and through the Curry-Howard correspondence we can turn it into a logical notion of linearity.

1.3 Linearity in logic

Classical sequent calculus (and intuitionistic sequent calculus as well) derives sequents of the form $\Gamma \vdash \Delta$ using left and right introduction rules for each logical connective, plus the special rules for axiom and cut. It contains particular rules for handling the sets of hypotheses and conclusions, known as the *weakening* and *contraction* rules:

$$\frac{\Gamma \vdash \Delta}{\Gamma, A \vdash \Delta}\ \text{wL} \qquad \frac{\Gamma \vdash \Delta}{\Gamma \vdash A, \Delta}\ \text{wR} \qquad \frac{\Gamma, A, A \vdash \Delta}{\Gamma, A \vdash \Delta}\ \text{cL} \qquad \frac{\Gamma \vdash A, A, \Delta}{\Gamma \vdash A, \Delta}\ \text{cR}$$

These rules allow each hypothesis to be used any number of times and it is crucial for the expressiveness of the logic. Note that, in the intuitionistic case (as in NJ) there is always one formula on the right of $\vdash$ hence weakening and contraction are relevant only in the left-hand side, i.e. the hypotheses.

The rest of classical sequent calculus has two presentations, depending on the treatment of the context of the active formula in each introduction rule. For instance, the right introduction rule for conjunction $\wedge$ has two flavours:

$$\frac{\Gamma \vdash A, \Delta \qquad \Gamma \vdash B, \Delta}{\Gamma \vdash A \wedge B, \Delta}\ \wedge\text{Ra} \qquad\qquad \frac{\Gamma \vdash A, \Delta \qquad \Gamma' \vdash B, \Delta'}{\Gamma, \Gamma' \vdash A \wedge B, \Delta, \Delta'}\ \wedge\text{Rm}$$

The first form is called *additive* and the second one is called *multiplicative*. For the left introduction rules, there are similarly two versions:

$$\frac{\Gamma, A \vdash \Delta}{\Gamma, A \wedge B \vdash \Delta}\ \wedge\text{La1} \qquad \frac{\Gamma, B \vdash \Delta}{\Gamma, A \wedge B \vdash \Delta}\ \wedge\text{La2} \qquad\qquad \frac{\Gamma, A, B \vdash \Delta}{\Gamma, A \wedge B \vdash \Delta}\ \wedge\text{Lm}$$

Similar variants exist for the other connectives. The multiplicative and additive variants of each rule are equivalent in the presence of weakening and contraction, however they have different properties when studying the structure of proofs.

In the intuitionistic case, weakening and contraction are not necessary if the system is presented in additive form, as in table 1. However, a multiplicative form could be defined, with explicit contraction and weakening, and it would bear more precise information about the handling of variables in λ-terms. Moreover, it is the key to the definition of logical linearity: a proof of $\Gamma, A \vdash B$ is linear in A if the A in conclusion is not involved in a contraction or weakening anywhere in the proof.

In some sense, the point of linear logic is to make this notion of linearity explicit in the language of formulas, and to refine logical deduction accordingly.

Structure

$$\frac{\Gamma, A, B, \Gamma' \vdash \Delta}{\Gamma, B, A, \Gamma' \vdash \Delta}\ \text{exL} \qquad \frac{\Gamma \vdash \Delta, A, B, \Delta'}{\Gamma \vdash \Delta, B, A, \Delta'}\ \text{exR}$$

Identity

$$\frac{}{A \vdash A}\ \text{ax} \qquad \frac{\Gamma \vdash \Theta, A, \Delta \qquad \Gamma', A, \Theta' \vdash \Delta'}{\Gamma, \Gamma', \Theta' \vdash \Theta, \Delta, \Delta'}\ \text{cut}$$

Negation

$$\frac{\Gamma, A \vdash \Delta}{\Gamma \vdash A^\perp, \Delta}\ \perp\text{R} \qquad \frac{\Gamma \vdash A, \Delta}{\Gamma, A^\perp \vdash \Delta}\ \perp\text{L}$$

Conjunction

$$\frac{\Gamma \vdash A, \Delta \qquad \Gamma' \vdash B, \Delta'}{\Gamma, \Gamma' \vdash A \otimes B, \Delta, \Delta'}\ \otimes\text{R} \qquad \frac{\Gamma, A, B \vdash \Delta}{\Gamma, A \otimes B \vdash \Delta}\ \otimes\text{L}$$

Disjunction

$$\frac{\Gamma \vdash A, B, \Delta}{\Gamma \vdash A \parr B, \Delta}\ \parr\text{R} \qquad \frac{\Gamma, A \vdash \Delta \qquad \Gamma', B \vdash \Delta'}{\Gamma, \Gamma', A \parr B \vdash \Delta, \Delta'}\ \parr\text{L}$$

Table 2: MLL sequent calculus in two-sided presentation

2 Linear sequent calculus

2.1 Multiplicative linear logic

We now define the smallest possible system in linear logic. It has only the multiplicative form of conjunction and disjunction and propositional variables as atomic formulas. Richer systems, including the additive form of connectives, units (the analogue of the *true* and *false* statements), quantifiers or modalities will be defined later. This minimal system is called MLL, for *multiplicative linear logic*.

12 **Definition**. The language of formulas of MLL is defined by the following grammar:

$$\begin{array}{lll} A, B := \alpha & & \text{propositional variable} \\ \quad A^\perp & & \text{linear negation — read “}A\text{ orthogonal” or “not }A\text{”} \\ \quad A \otimes B & & \text{multiplicative conjunction — read “}A\text{ tensor }B\text{”} \\ \quad A \parr B & & \text{multiplicative disjunction — read “}A\text{ par }B\text{”} \end{array}$$

13 **Definition**. A two-sided MLL sequent is a pair (Γ, Δ) of possibly empty lists of MLL formulas. Such a pair is usually written $\Gamma \vdash \Delta$ and read “Γ *entails* Δ”.

14 **Definition**. A sequential[2] two-sided MLL proof is a proof tree built using the rules of table 2. A sequent $\Gamma \vdash \Delta$ is *provable* in MLL if there exists some proof π whose conclusion is $\Gamma \vdash \Delta$; we write this fact as $\pi : \Gamma \vdash \Delta$. By extension, we say that a formula A is provable if the sequent $\vdash A$ is provable.

The *exchange rules* (“exL” on the left and “exR” on the right) state that the order in which the formulas occur in the sequents is irrelevant, only the side on which they occur matters. The *axiom rule*,

[2]The word “sequential” conveniently refers to *sequent* calculus, but it is also used in contrast to proof nets, defined below in section 5, which are more *parallel* in nature.

abbreviated as "ax", is in multiplicative form, with an empty context on both sides. The *cut rule* is also multiplicative: contexts from both premises are concatenated. All other rules are *introduction rules* for each connective, they have a name that ends with "L" for left rules or "R" for right rules, depending on the side of the entailment symbol $\vdash$ on which they introduce a formula in conclusion.

15 *Example.* The following proof establishes the commutativity of multiplicative conjunction:

$$\dfrac{\dfrac{\dfrac{\dfrac{}{B \vdash B}\,\text{ax} \qquad \dfrac{}{A \vdash A}\,\text{ax}}{B, A \vdash B \otimes A}\,\otimes\text{R}}{A, B \vdash B \otimes A}\,\text{exL}}{A \otimes B \vdash B \otimes A}\,\otimes\text{L}$$

16 **Exercise**. Prove that the following sequents are provable in MLL:

- multiplicative excluded middle: $\vdash A \parr A^\perp$
- semi-distributivity of tensor over par: $A \otimes (B \parr C) \vdash (A \otimes B) \parr C$

17 **Proposition**. *Let $\Gamma \vdash \Delta$ be an MLL sequent. An occurrence of a propositional variable α in $\Gamma \vdash \Delta$ is called* positive *if it occurs under an even number of negations in Δ or under an odd number of negations in Γ. If $\Gamma \vdash \Delta$ is provable in MLL, then each propositional variable has an equal number of positive and negative occurrences.*

Proof. It is clear that this property holds in the conclusion of an axiom rule and that it is preserved by all other logical rules. □

This linearity property allows to easily recognize as not provable any formula that violates this condition, for instance $A^\perp \parr (A \otimes A)$ is not provable in MLL.

18 **Definition**. Two formulas A and B are *linearly equivalent* if the sequents $A \vdash B$ and $B \vdash A$ are provable in MLL. This fact is written $A \multimap\!\!\circ\, B$.

19 *Example.* In the proof of example 15, if we replace A with B and vice-versa, we get a proof of $B \otimes A \vdash A \otimes B$. As a consequence, $A \otimes B$ and $B \otimes A$ are linearly equivalent.

20 **Exercise**. Prove that the linear equivalence of two formulas A and B is equivalent to the fact that, for all lists of formulas Γ and Δ, the sequent $\Gamma \vdash A, \Delta$ is provable if and only if the sequent $\Gamma \vdash B, \Delta$ is provable.

21 **Exercise**. Define the system MLL′ as the logical system MLL extended with a new connective $A \multimap B$ called *linear implication* and read "A implies B". The system of inference rules is extended with the introduction rules for this new connective:

$$\dfrac{\Gamma, A \vdash B, \Delta}{\Gamma \vdash A \multimap B, \Delta}\,\multimap\text{R} \qquad\qquad \dfrac{\Gamma \vdash A, \Delta \qquad \Gamma', B \vdash \Delta'}{\Gamma, \Gamma', A \multimap B \vdash \Delta, \Delta'}\,\multimap\text{L}$$

Prove that, in this system, for any A and B, the formulas $A \multimap B$ and $A^\perp \parr B$ are linearly equivalent.

22 As consequence of this fact, in MLL, it is customary to not consider linear implication $A \multimap B$ as a proper connective, but merely as a notation for $A^\perp \parr B$. Nevertheless, this notation is widely used since it is very meaningful with respect to the proof system, because it carries essentially the same logical meaning as the entailment symbol $\vdash$. Indeed, the sequent $A \vdash B$ is provable if and only if the formula $A \multimap B$ is provable. This also justifies the notation $A \multimap\!\!\circ\, B$ for linear equivalence.

2.2 Cut elimination and consistency

Sequent calculus was originally introduced by Gentzen in order to prove the consistency or arithmetic, and this method was later adapted to many logical systems as a standard way to establish consistency results. The main lemma for consistency, sometimes called *Hauptsatz* (which means *main result* in German), is the *cut elimination* property, stating that any proof can be transformed into a proof that does not use the cut rule.[3] Linear logic also enjoys this property, and we will prove it now for MLL.

The idea is to start from an arbitrary proof with cut rules and to rewrite each instance of the cut rule in order to push it upwards to the axiom rules, eventually eliminating all cuts.

We are (for now) working with a very strict system in which reordering the hypotheses and conclusions of a proof takes explicit operation using exchange rules. The left and right exchange rules allow single transpositions of formulas on either side of a sequent; since transpositions generate all permutations, it is clear that any permutation of the formulas in a sequent can be implemented by some sequence of left and right exchange rules. We will write

$$\frac{\Gamma \vdash \Delta}{\Gamma' \vdash \Delta'}\ \mathrm{ex}$$

for any such sequence of rules, assuming Γ' is a permutation of Γ and Δ' is a permutation of Δ, when the precise order of these exchange rules is irrelevant.

23 **Definition**. The cut elimination relation $\rightsquigarrow$ is the congruent binary relation over proofs generated by the following rules:

- If the active formula is in the context of the last rule of one of the premisses, then this last rule is exchanged with the cut rule. For instance, if the left premiss ends with a right tensor rule and the active formula is in the context of the left premiss of this rule, we have

$$\dfrac{\dfrac{\pi_{1b} : \Gamma_{1b} \vdash B, \Theta_1, A, \Delta_{1b} \qquad \pi_{1c} : \Gamma_{1c} \vdash C, \Delta_{1c}}{\Gamma_{1b}, \Gamma_{1c} \vdash B \otimes C, \Theta_1, A, \Delta_{1b}, \Delta_{1c}}\otimes\mathrm{R} \qquad \pi_2 : \Gamma_2, A, \Theta_2 \vdash \Delta_2}{\Gamma_{1b}, \Gamma_{1c}, \Gamma_2, \Theta_2 \vdash B \otimes C, \Theta_1, \Delta_{1b}, \Delta_{1c}, \Delta_2}\ \mathrm{cut}$$

$$\rightsquigarrow$$

$$\dfrac{\dfrac{\pi_{1b} : \Gamma_{1b} \vdash B, \Theta_1, A, \Delta_{1b} \qquad \pi_2 : \Gamma_2, A, \Theta_2 \vdash \Delta_2}{\Gamma_{1b}, \Gamma_2, \Theta_2 \vdash B, \Theta_1, \Delta_{1b}, \Delta_2}\ \mathrm{cut} \qquad \pi_{1c} : \Gamma_{1c} \vdash C, \Delta_{1c}}{\dfrac{\Gamma_{1b}, \Gamma_2, \Theta_2, \Gamma_{1c} \vdash B \otimes C, \Theta_1, \Delta_{1b}, \Delta_2, \Delta_{1c}}{\Gamma_{1b}, \Gamma_{1c}, \Gamma_2, \Theta_2 \vdash B \otimes C, \Theta_1, \Delta_{1b}, \Delta_{1c}, \Delta_2}\ \mathrm{ex}}\otimes\mathrm{R}$$

  The other cases are similar, we allow such commutations for any rule in the premisses, including the cut rule itself.

- If the active formula is introduced by an exchange rule in one of the premisses of the cut, then this exchange rule may be dropped. An instance of this is

$$\dfrac{\dfrac{\pi_1 : \Gamma_1 \vdash \Theta_1, A, B, \Delta_1}{\Gamma_1 \vdash \Theta_1, B, A, \Delta_1}\ \mathrm{exR} \qquad \pi_2 : \Gamma_2, A, \Theta_2 \vdash \Delta_2}{\Gamma_1, \Gamma_2, \Theta_2 \vdash \Theta_1, B, \Delta_1, \Delta_2}\ \mathrm{cut}$$

$$\rightsquigarrow \dfrac{\pi_1 : \Gamma_1 \vdash \Theta_1, A, B, \Delta_1 \qquad \pi_2 : \Gamma_2, A, \Theta_2 \vdash \Delta_2}{\Gamma_1, \Gamma_2, \Theta_2 \vdash \Theta_1, B, \Delta_1, \Delta_2}\ \mathrm{cut}$$

[3] Of course, by proposition 17, we already know that MLL is not degenerate, since any sequent that does not respect linearity is not provable. However, cut elimination is by far more general and it carries the actual meaning of the proof system, contrary to the linearity property, which is nothing more than a secondary observation.

- If the active formula is introduced by an axiom rule in one of the premisses of the cut, then this axiom rule and the cut itself may be dropped, with some exchanges possibly introduced to preserve the order of conclusions. An instance of this is

$$\dfrac{\dfrac{}{A \vdash A}\,\text{ax} \qquad \pi_2 : \Gamma_2, A, \Theta_2 \vdash \Delta_2}{A, \Gamma_2, \Theta_2 \vdash \Delta_2}\,\text{cut} \rightsquigarrow \dfrac{\pi_2 : \Gamma_2, A, \Theta_2 \vdash \Delta_2}{A, \Gamma_2, \Theta_2 \vdash \Delta_2}\,\text{ex}$$

- If the active formula is introduced on both sides of the cut by the introduction rule for some connective, then we have a right introduction rule on the left and an left introduction rule on the right for the same connective. In this case, we can eliminate these introduction rules and the cut rule and introduce instead a cut rule for each immediate subformula.

  For negation we have:

$$\dfrac{\dfrac{\pi_1 : \Gamma_1, B \vdash \Delta_1}{\Gamma_1 \vdash B^\perp, \Delta_1}\,\perp\text{R} \qquad \dfrac{\pi_2 : \Gamma_2 \vdash B, \Delta_2}{\Gamma_2, B^\perp \vdash \Delta_2}\,\perp\text{L}}{\Gamma_1, \Gamma_2 \vdash \Delta_1, \Delta_2}\,\text{cut} \rightsquigarrow \dfrac{\dfrac{\pi_2 : \Gamma_2 \vdash B, \Delta_2 \qquad \pi_1 : \Gamma_1, B \vdash \Delta_1}{\Gamma_2, \Gamma_1 \vdash \Delta_2, \Delta_1}\,\text{cut}}{\Gamma_1, \Gamma_2 \vdash \Delta_1, \Delta_2}\,\text{ex}$$

  For the tensor we have:

$$\dfrac{\dfrac{\pi_{1b} : \Gamma_{1b} \vdash B, \Delta_{1b} \qquad \pi_{1c} : \Gamma_{1c} \vdash C, \Delta_{1c}}{\Gamma_{1b}, \Gamma_{1c} \vdash B \otimes C, \Delta_{1b}, \Delta_{1c}}\,\otimes\text{R} \qquad \dfrac{\pi_2 : \Gamma_2, B, C \vdash \Delta_2}{\Gamma_2, B \otimes C \vdash, \Delta_2}\,\otimes\text{L}}{\Gamma_{1b}, \Gamma_{1c}, \Gamma_2 \vdash \Delta_{1b}, \Delta_{1c}, \Delta_2}\,\text{cut}$$

$$\rightsquigarrow$$

$$\dfrac{\pi_{1b} : \Gamma_{1b} \vdash B, \Delta_{1b} \qquad \dfrac{\pi_{1c} : \Gamma_{1c} \vdash C, \Delta_{1c} \qquad \pi_2 : \Gamma_2, B, C \vdash \Delta_2}{\Gamma_{1c}, \Gamma_2, B \vdash \Delta_{1c}, \Delta_2}\,\text{cut}}{\Gamma_{1b}, \Gamma_{1c}, \Gamma_2 \vdash \Delta_{1b}, \Delta_{1c}, \Delta_2}\,\text{cut}$$

  For the par we have:

$$\dfrac{\dfrac{\pi_1 : \Gamma_1 \vdash B, C, \Delta_1}{\Gamma_1 \vdash B \parr C, \Delta_1}\,\parr\text{R} \qquad \dfrac{\pi_{2b} : \Gamma_{2b}, B \vdash \Delta_{2b} \qquad \pi_{2c} : \Gamma_{2c}, C \vdash \Delta_{2c}}{\Gamma_{2b}, \Gamma_{2c}, B \parr C \vdash \Delta_{2b}, \Delta_{2c}}\,\parr\text{L}}{\Gamma_1, \Gamma_{2b}, \Gamma_{2c} \vdash \Delta_1, \Delta_{2b}, \Delta_{2c}}\,\text{cut}$$

$$\rightsquigarrow$$

$$\dfrac{\dfrac{\pi_1 : \Gamma_1 \vdash B, C, \Delta_1 \qquad \pi_{2b} : \Gamma_{2b}, B \vdash \Delta_{2b}}{\Gamma_1, \Gamma_{2b} \vdash C, \Delta_1, \Delta_{2b}}\,\text{cut} \qquad \pi_{2c} : \Gamma_{2c}, C \vdash \Delta_{2c}}{\Gamma_1, \Gamma_{2b}, \Gamma_{2c} \vdash \Delta_1, \Delta_{2b}, \Delta_{2c}}\,\text{cut}$$

  These three cases are called *interaction* rules.

It is easy to check that for every instance of the cut rule in some proof, at least one of the reduction rules applies. Therefore, a proof that is irreducible for this relation $\rightsquigarrow$ contains no instance of the cut rule. Besides, by construction, if $\pi \rightsquigarrow \pi'$ then π and π' have the same conclusion. The proof of the cut elimination property thus consists in proving that any proof can be reduced into an irreducible one.

24 **Lemma**. *Every sequence of cut elimination steps that never commutes a cut rule with another cut rule is finite.*

Proof. We use a termination measure to prove this fact. To every instance of the cut rule

$$\frac{\pi_1 : \Gamma_1 \vdash \Theta_1, A, \Delta_1 \qquad \pi_2 : \Gamma_2, A, \Theta_2 \vdash \Delta_2}{\Gamma_1, \Gamma_2, \Theta_2 \vdash \Theta_1, \Delta_1, \Delta_2}\ \text{cut}$$

we associate the pair $(h(A), h(\pi_1)+h(\pi_2))$ where h is the height of a formula or proof as a tree. This pair is called the *measure* of this instance of the cut rule. Measures are compared by lexicographic ordering. It is easy to check that, in each rule in the definition or $\rightsquigarrow$, the cuts introduced in the right-hand side have a strictly smaller measure than the eliminated cut on the left-hand side. Indeed, in the case of interaction rules, the new cuts apply to subformulas of the original formula, so $h(A)$ decreases strictly. In all other cases, the new cuts are applied to subproofs of the premisses of the original cut, so $h(A)$ is unchanged and $h(\pi_1) + h(\pi_2)$ decreases strictly.

Cut measures take values in $\mathbb{N} \times \mathbb{N}$ with the lexicographic ordering, which is well-founded. If the measure of a proof π is defined as the multiset $m(\pi)$ of the measures of the instances of the cut rule in π, then we deduce that $\pi \rightsquigarrow \pi'$ implies $m(\pi) > m(\pi')$. The multiset ordering over a well-founded set is itself well-founded, hence there can be no infinite sequence of cut elimination steps starting from an MLL proof. □

Note that it is crucial in this lemma that the rule for exchanging two cuts is never used, otherwise one could trivially form an infinite sequence by repeatedly commuting the same pair of cut rules.

25 **Theorem**. *A sequent $\Gamma \vdash \Delta$ is provable in MLL if and only if it has a proof that does not use the cut rule.*

Proof. Assume $\Gamma \vdash \Delta$ is provable in MLL. Consider a proof π_0 of $\Gamma \vdash \Delta$ and a maximal sequence $\pi_0 \rightsquigarrow \pi_1 \rightsquigarrow \dots \rightsquigarrow \pi_n \rightsquigarrow \dots$ that never commutes two cut rules. By definition of $\rightsquigarrow$, each proof π_i has conclusion $\Gamma \vdash \Delta$. By lemma 24, this sequence must be finite. Its last element is thus a proof of $\Gamma \vdash \Delta$ that is irreducible by the relation $\rightsquigarrow$. This implies that this last proof contains no occurrence of the cut rule. □

Another way of formulating this result is saying that the cut rule is *admissible* in the proof system MLL without cut. This means that the cut rule does not affect the expressiveness of the system in terms of provability. This fact is crucial for establishing consistency because it immediately implies that not all sequents are provable, so the proof system is not trivial.

26 **Corollary**. *The empty sequent $\vdash$ is not provable in MLL.*

Proof. Every rule of MLL except the cut rule has at least one formula in its conclusion, on one side of the sequent or the other. Therefore no cut-free proof can have the empty sequent as conclusion. By theorem 25, this implies that $\vdash$ is not provable. □

To sum up, the idea of this proof is to define a computational procedure, the cut-elimination relation $\rightsquigarrow$, that transforms an arbitrary proof into a cut-free proof with the same conclusion. The fact that this relation is well-founded proves that the procedure always terminates, hence all proofs have a cut-free form. This notion of proof rewriting is the core of the proof-program correspondence and we will elaborate later on the computational part of this correspondence for MLL in particular.

2.3 One-sided presentation

In the rules of table 2, it appears that the connectives $\otimes$ and $\parr$ are very symmetric: they have the same rules, only with the left and right sides of sequents permuted. The cut elimination rules of definition 23 are consistent with this and, indeed, the cut elimination procedure behaves in a very similar way for both connectives. Everything that happens on the left for $\otimes$ happens on the right for $\parr$, and conversely. This is a strengthened form of the De Morgan duality, which is well-known in classical logic.

Identity

$$\frac{}{\vdash A^\perp, A}\ \text{ax} \qquad\qquad \frac{\vdash \Gamma, A \qquad \vdash A^\perp, \Delta}{\vdash \Gamma, \Delta}\ \text{cut}$$

Multiplicatives

$$\frac{\vdash \Gamma, A \qquad \vdash B, \Delta}{\vdash \Gamma, A \otimes B, \Delta}\ \otimes \qquad\qquad \frac{\vdash \Gamma, A, B}{\vdash \Gamma, A \parr B}\ \parr$$

Table 3: MLL sequent calculus in one-sided presentation

27 **Exercise (De Morgan laws).** Prove that, for all formulas A and B, the following linear equivalences hold:

$$A \multimap A^{\perp\perp}, \qquad (A \otimes B)^\perp \multimap A^\perp \parr B^\perp, \qquad (A \parr B)^\perp \multimap A^\perp \otimes B^\perp.$$

This property is the exact analogue of the De Morgan laws of classical logic, that state the duality between conjunction and disjunction: $\neg(A \wedge B)$ is equivalent to $\neg A \vee \neg B$ and conversely. If we apply De Morgan laws recursively in subformulas, we can easily establish that any formula of MLL is linearly equivalent to a formula where negation is only applied to propositional variables. Moreover, the introduction rules for negation imply that a sequent $\Gamma, A \vdash \Delta$ is provable in MLL if and only if the sequent $\Gamma \vdash A^\perp, \Delta$ is provable.

These considerations motivate the introduction of the *one-sided* presentation of MLL. The idea is to combine the above remarks so that

- formulas are considered up to De Morgan equivalences,
- all sequents have the form $\vdash \Gamma$, with an empty left-hand side,
- subsequently, only right introduction rules are used.

Moreover, sequents will be considered up to permutation, since exchange rules make permutation of formula instances innocuous.

28 **Definition**. The language of formulas of one-sided MLL is defined by the following grammar:

$$\begin{array}{rll} A, B := & \alpha & \text{propositional variable} \\ & \alpha^\perp & \text{negated propositional variable} \\ & A \otimes B & \text{multiplicative conjunction} \\ & A \parr B & \text{multiplicative disjunction} \end{array}$$

Linear negation is the operation $(\cdot)^\perp$ on formulas inductively defined as

$$(\alpha)^\perp := \alpha^\perp, \qquad (\alpha^\perp)^\perp := \alpha, \qquad (A \otimes B)^\perp := A^\perp \parr B^\perp, \qquad (A \parr B)^\perp := A^\perp \otimes B^\perp.$$

A one-sided MLL sequent is a possibly empty multiset Γ of one-sided MLL formulas, usually written as $\vdash \Gamma$. A sequential one-sided MLL proof is a proof tree built using the rules of table 3.

Note that the axiom and cut rules have to be reformulated to fit in this context. As expected, the system is greatly simplified since we now have only four rules, instead of ten in the one-sided presentation. Expressiveness is preserved, as the following theorem states.

29 **Theorem**. *A sequent $A_1, ..., A_n \vdash B_1, ..., B_p$ is provable in two-sided MLL if and only if the sequent $\vdash A_1^\perp, ..., A_n^\perp, B_1, ..., B_p$ is provable in one-sided MLL.*

Proof. Each proof π of two-sided MLL with some conclusion $\Gamma \vdash \Delta$ can be transformed into a proof π' of $\vdash \Gamma^\perp, \Delta$, and this transformation is defined inductively, case by case on the last rule:

- if π is a two-sided axiom, then the one-sided axiom rule is a suitable π',
- if π is a two-sided cut between π_1 and π_2, then the one-sided cut between π'_1 and π'_2 is a suitable π',
- right introduction rules are translated into introduction rules for the same connectives,
- left introduction rules are translated into introduction rules for the dual connectives,
- introduction rules for negation and exchange rules have no effect on the conclusion in the one-sided form so they disappear in π'.

For the reverse implication, each rule of one-sided MLL is mapped to the rule of two-sided MLL with the same shape. Introductions are translated back into left or right introduction rules depending on whether the introduced formula is an A_i or a B_j. Exchange rules and negation introductions are inserted as needed for the rules to apply. □

Consistency of one-sided MLL is a consequence of this theorem. However, it could be established directly, using the same technique based on proof rewriting, and the cut-elimination procedure is actually simpler because there are much less commutation rules and only one interaction rule, for the pair $\otimes/\parr$. Besides, the cut elimination relation $\rightsquigarrow$ of definition 23 is mapped by the proof translation of theorem 29 to the appropriate cut elimination relation for one-sided MLL.

2.4 Full linear logic

So far, we have focused on the very simple MLL system, which contains only multiplicative forms of conjunction and disjunction. The one-sided presentation, which includes "hard-wired" De Morgan duality, provides a simple sequent-style proof system with good properties, so we will keep using one-sided presentations whenever possible.

The full system of linear logic is obtained by introducing in the system all the other connectives needed to get back the expressiveness of full classical logic. We will now briefly describe all these, together with their cut elimination procedure.

Additives

The additive form of conjunction is written & and pronounced "with", the additive form of disjunction is written $\oplus$ and pronounced "plus". They are dual, in the sens that the definition of negation is extended with the rules

$$(A \mathbin{\&} B)^\perp := A^\perp \oplus B^\perp \qquad \text{and} \qquad (A \oplus B)^\perp := A^\perp \mathbin{\&} B^\perp.$$

Their proof rules are

$$\frac{\vdash \Gamma, A \qquad \vdash \Gamma, B}{\vdash \Gamma, A \mathbin{\&} B}\ \& \qquad \frac{\vdash \Gamma, A}{\vdash \Gamma, A \oplus B}\ \oplus_1 \qquad \frac{\vdash \Gamma, B}{\vdash \Gamma, A \oplus B}\ \oplus_2$$

The names $\oplus_1$ and $\oplus_2$ refer to the fact that the rules are very similar, to the point that they may be formulated as a single rule:

$$\frac{\vdash \Gamma, A_i}{\vdash \Gamma, A_1 \oplus A_2}\ \oplus_i$$

where i is either 1 or 2. Their treatment in cut elimination is similar to the one for multiplicative connectives. There are now two interaction rules, for & against $\oplus_1$ and for & against $\oplus_2$, and they have a very similar shape:

$$\dfrac{\dfrac{\pi_{1,1} \vdash \Gamma, A_1 \qquad \pi_{1,2} \vdash \Gamma, A_2}{\vdash \Gamma, A_1 \,\&\, A_2}\,\& \qquad \dfrac{\pi_2 \vdash \Delta, A_i^\perp}{\vdash \Delta, A_1^\perp \oplus A_2^\perp}\,\oplus_i}{\vdash \Gamma, \Delta}\,\text{cut} \quad \rightsquigarrow \quad \dfrac{\pi_{1,i} \vdash \Gamma, A_i \qquad \pi_2 \vdash \Delta, A_i^\perp}{\vdash \Gamma, \Delta}\,\text{cut}$$

Remark that one of the premisses of the & rule is erased. As for the commutation rules, the case of a cut against a formula in the context of one of the additive rules is treated in the only possible way. Commutation with $\oplus_i$ is straightforward and commutation with & implies duplicating the other proof:

$$\dfrac{\dfrac{\pi_{1,1} \vdash \Gamma, A, B_1 \qquad \pi_{1,2} \vdash \Gamma, A, B_2}{\vdash \Gamma, A, B_1 \,\&\, B_2}\,\& \qquad \pi_2 \vdash \Delta, A^\perp}{\vdash \Gamma, \Delta, B_1 \,\&\, B_2}\,\text{cut}$$

$$\rightsquigarrow$$

$$\dfrac{\dfrac{\pi_{1,1} \vdash \Gamma, A, B_1 \qquad \pi_2 \vdash \Delta, A^\perp}{\vdash \Gamma, \Delta, B_1}\,\text{cut} \qquad \dfrac{\pi_{1,2} \vdash \Gamma, A, B_2 \qquad \pi_2 \vdash \Delta, A^\perp}{\vdash \Gamma, \Delta, B_2}\,\text{cut}}{\vdash \Gamma, \Delta, B_1 \,\&\, B_2}\,\&$$

The termination measure used in lemma 24 does not work directly, because this rule might duplicate cuts in π_2 that are bigger than the cut it reduces. However, if we restrict this rule so that it only applies when cuts in π_2 are small enough (at worst, apply it only if π_2 is already cut-free!), then we get a reduction strategy that reaches a normal form, so consistency is preserved. The logical system we obtain is called *multiplicative-additive linear logic* and abbreviated as MALL.

30 We have the following notations for the connectives:

	conjunction	disjunction
multiplicative	$\otimes$	$\parr$
additive	&	$\oplus$

This may look like a strange decision to use these symbols since connectives of the same kind do not look alike. One of the justifications of this choice is that connectives that are similar in appearance interact nicely together.

31 **Exercise**. Prove the following linear equivalences in MALL:

$$A \otimes (B \oplus C) \multimap (A \otimes B) \oplus (A \otimes C) \qquad \text{and} \qquad A \parr (B \,\&\, C) \multimap (A \parr B) \,\&\, (A \parr C).$$

These distributivity laws are similar to the standard calculus laws of distribution of multiplication over addition, the similar shape of the connectives help in remembering which distributivities hold. Indeed, $A \parr (B \oplus C)$ is not linearly equivalent to $(A \parr B) \oplus (A \parr C)$.

32 **Exercise**. Prove that.

In the computational interpretation of proofs deduced from the cut elimination procedure, we interpret a proof of $A \vdash B$ (or equivalently a proof of $\vdash A^\perp, B$) as a way to map a proof of A to a proof of B. The cut elimination procedure defines how a given proof of A is *used* to produce the resulting proof of B. Comparing the cut elimination behaviour of multiplicative and additive connectives illustrates the difference in meaning between them. Assuming δ (the data) is a proof of $\vdash A * B$ for some connective $*$ and φ (the function) is a proof of $A * B \vdash C$, what will happen when performing cut elimination in the cut of δ against φ?

- for $A \oplus B$, δ provides *either* a proof of A *or* a proof of B;
- for $A \mathbin{\&} B$, δ provides a proof of A *and* a proof of B but φ must use *one* of them;
- for $A \otimes B$, δ provides a proof of A *and* a proof of B and φ must use *both* of them;
- for $A \parr B$, δ provides one proof with two conclusions A and B, and φ must use them *independently*, with some subproof for A and some other subproof for B.

Units

In classical logic, there are two logical units, standing for *true* and *false*. A way of characterizing them is as neutral elements for conjunction and disjunction, respectively. For this reason, in linear logic, there are multiplicative and additive variants of these units, hence there are four units. Their notations follow the pattern of the conjunctions and disjunctions:

	"true"	"false"
multiplicative	1	$\bot$
additive	$\top$	0

The definition of negation extends in the natural way:

$$1^\bot := \bot, \qquad \bot^\bot := 1, \qquad \top^\bot := 0, \qquad 0^\bot := \top.$$

The proof rules are deduced from the neutrality property:

$$\frac{}{\vdash 1}\ 1 \qquad \frac{\vdash \Gamma}{\vdash \Gamma, \bot}\ \bot \qquad \frac{}{\vdash \Gamma, \top}\ \top$$

33 There is no rule for 0. This can be justified by the computational interpretation and the neutrality for $\oplus$: a proof of $A \oplus B$ is either a proof of A or a proof of B, so the set of proofs of $A \oplus B$ is mostly the disjoint union of the set of proofs of A and the set of proofs of B. But if $A \oplus 0$ and A must be equivalent, this means that the set of proofs of $A \oplus 0$ and the set of proofs of A are essentially the same, therefore there must be no proof of 0.

The cut elimination rules are extended in the straightforward way: interaction of 1 against $\bot$ simply cancels both rules, interaction between $\top$ and 0 never happens since there is no rule for 0. The context rule for $\bot$ is a simple commutation and the context rule for $\top$ erases the cut and its other premiss:

$$\dfrac{\dfrac{}{\vdash \Gamma, A, \top}\ \top \qquad \pi \vdash \Delta, A^\bot}{\vdash \Gamma, \Delta, \top}\ \mathrm{cut} \quad \rightsquigarrow \quad \frac{}{\vdash \Gamma, \Delta, \top}\ \top$$

This is consistent with the context rule for &.

34 **Exercise**. Prove the following linear equivalences:

- neutralities: $A \multimap A \otimes 1 \multimap A \parr \bot \multimap A \mathbin{\&} \top \multimap A \oplus 0$,
- nullity: $A \otimes 0 \multimap 0$, $A \parr \top \multimap \top$.

Exponentials

Limiting the logical system to multiplicative and additive variants of classical conjunction and disjunction, without any contraction and weakening, does provide an interesting system, however its logical expressiveness is rather limited. A witness of this fact is that the height of cut-free proofs of a given sequent is obviously bounded by the number of connectives in the sequent, since each rule introduces one connective. As a consequence, proof search in MALL can be performed by exhaustive enumeration of all possible proofs and the decision problem of finding whether a given sequent is provable is decidable.[4]

In order to recover the expressiveness of classical logic, it is thus necessary to allow some contraction and weakening. Introducing them directly would collapse the system into classical logic itself, so we introduce them in a controlled way, by means of *modalities*. The language of formulas is extended with a pair of new constructs:

$$\begin{aligned} A, B &:= \dots \\ &\quad !A \quad \text{replicable and erasable — read “of course } A\text{”} \\ &\quad ?A \quad \text{possibly contracted or weakened — read “why not } A\text{”} \end{aligned}$$

De Morgan duality is extended so that these are dual: $(!A)^\perp := ?(A^\perp)$ and $(?A)^\perp := !(A^\perp)$. The modality ? is a marker for formulas that can be contracted and weakened:

$$\frac{\vdash \Gamma, A}{\vdash \Gamma, ?A}\ ? \qquad \frac{\vdash \Gamma}{\vdash \Gamma, ?A}\ \mathrm{w} \qquad \frac{\vdash \Gamma, ?A, ?A}{\vdash \Gamma, ?A}\ \mathrm{c} \qquad \frac{\vdash ?A_1, \dots ?A_n, B}{\vdash ?A_1, \dots ?A_n, !B}\ !$$

The introduction rule for ? is usually called “dereliction”, as it is a kind of regression: the linearity information is lost. The introduction rule for !, usually called “promotion”, is very particular since it imposes a constraint on the context, indeed if we use the proof with conclusion B several times later in our proof, we will have to use the other conclusions the same number of times, hence these cannot be linear.

The cut elimination rules naturally imply duplicating or erasing the proof above a promotion rule. For instance, when a cut occurs between a contraction and a promotion, we have

$$\frac{\dfrac{\pi_1 \vdash \Gamma, ?A, ?A}{\vdash \Gamma, ?A}\ \mathrm{c} \qquad \dfrac{\pi_2 \vdash ?\Delta, A^\perp}{\vdash ?\Delta, !A^\perp}\ !}{\vdash \Gamma, ?\Delta}\ \mathrm{cut} \quad \rightsquigarrow \quad \frac{\dfrac{\pi_1 \vdash \Gamma, ?A, ?A \qquad \dfrac{\pi_2 \vdash ?\Delta, A^\perp}{\vdash ?\Delta, !A^\perp}\ !}{\vdash \Gamma, ?\Delta, ?A}\ \mathrm{cut} \qquad \dfrac{\pi_2 \vdash ?\Delta, A^\perp}{\vdash ?\Delta, !A^\perp}\ !}{\dfrac{\vdash \Gamma, ?\Delta, ?\Delta}{\vdash \Gamma, ?\Delta}\ \mathrm{c}}\ \mathrm{cut}$$

where $?\Delta$ stands for a sequence of formulas all starting with the ? modality, and where the right-hand side ends with as many instances of the contraction rule as necessary to get the appropriate conclusion. Interaction with weakening is similar except that the promotion is erased and not duplicated, interaction with dereliction is a simple cancellation. Commutation rules can be formulated too, paying attention to the contextual constraint in the promotion rule. The termination argument of lemma 24 does not work anymore in this context, because of duplication: the rule above does not decrease this measure. This can be fixed by taking into account the number of contraction rules applied to the active formula in each cut, which does decrease.

These modalities are called *exponential* because they have the fundamental property of turning additives into multiplicatives.

[4] The problem of deciding the provability of an arbitrary MALL sequent was actually proved to be complete for polynomial space complexity, whereas the same problem for full propositional linear logic is undecidable. See Lincoln et al., “Decision problems for propositional linear logic”.

35 **Exercise**. Prove the linear equivalences

$$!(A \,\&\, B) \multimap\!\!\circ\; !A \otimes !B, \qquad ?(A \oplus B) \multimap\!\!\circ\; ?A \parr ?B, \qquad !\top \multimap\!\!\circ\; 1, \qquad ?0 \multimap\!\!\circ\; \bot.$$

Quantifiers

Our exposition so far has been restricted to propositional logic, since it is already very expressive in linear logic. However, it is possible to include quantifiers and get the full mathematical expressiveness of traditional predicate calculus, in a refined way.

First-order quantifiers are introduced in the standard way: choose a language for first-order objects (variables, function symbols, predicate symbols), replace propositional variables by atomic formulas made of a predicate symbol applied to the right number of first-order terms, and extend the language of formulas with the usual quantifiers $\forall x$ and $\exists x$, dual of each other: $(\forall x\, A)^\perp = \exists x (A^\perp)$. Proof rules and cut elimination steps are the same as in classical logic, cut elimination keeps the same complexity, and on the whole, the proof theory is not very much affected by this change.

Second-order quantifiers are more interesting from the point of view of the structure of proofs. Now we extend the language of formulas with quantification over propositional variables $\forall\alpha$ and $\exists\alpha$. These quantifiers are also dual of each other, the proof rules and cut elimination rules are as simple as for first order, even more so since the language is simpler. The real difference lies in the argument for termination of cut elimination: now we cannot proceed by induction on formulas anymore, since substitution of propositional variables make formulas grow. It is now necessary to use arguments comparable to reducibility candidates (as used for second-order intuitionistic natural deduction, a.k.a. system F, a.k.a. polymorphic λ-calculus).

An interesting side-effect of second-order quantification is that they make additives and units unnecessary, with respect to linear equivalence:

$$0 \multimap\!\!\circ\; \forall\alpha\, \alpha \qquad 1 \multimap\!\!\circ\; \forall\alpha(\alpha \multimap \alpha) \qquad A \oplus B \multimap\!\!\circ\; \forall\alpha(!(A \multimap \alpha) \multimap !(B \multimap \alpha) \multimap \alpha)$$
$$\top \multimap\!\!\circ\; \exists\alpha\, \alpha \qquad \bot \multimap\!\!\circ\; \exists\alpha(\alpha \otimes \alpha^\perp) \qquad A \,\&\, B \multimap\!\!\circ\; \exists\alpha(!(A \parr \alpha) \otimes !(B \parr \alpha) \otimes \alpha^\perp)$$

The semantic study of quantifiers in linear logic is rich topic with many open questions, and we will not elaborate on this in the present notes. Besides, the propositional part already illustrates most aspects of the proof theory of linear logic.

2.5 The notion of fragment

As already mentioned before, it is often useful to study subsystems of linear logic. These are often simpler because they have less connectives, less proof rules and less cut-elimination rules. They may still be expressive enough to represent some other systems or to illustrate particular features. For these reasons, many works focus on various subsystems, some of which have become standard systems in their own right.

A *fragment* of linear logic is a system obtained by imposing restrictions on the language of formulas, sequents or proofs. The most standard ones, based on the restriction of the set of formulas, are propositional fragments, with no quantifiers of any order and no units:

- LL: all connectives allowed
- MLL: only multiplicatives
- MALL: only multiplicatives and additives
- MELL: only multiplicatives and exponential modalities — this is enough to represent and study the simply typed λ-calculus and has been used extensively for this purpose

In these systems, the rules that involve forbidden connectives are simply never used. The same systems extended with the units are named similarly with a 0 as subscript. The systems with quantifiers are named with a 1 or 2 (or both) as subscripts, for first and second order quantification. With these conventions, MLL_{02} allows $\otimes, \parr, 1, \bot$ and second-order quantification; MALL_1 allows $\otimes, \parr, \oplus, \&$ and first-order quantification but no units, and LL_{012} is the full system.

Fragments can also be defined by restricting the shape of sequents. The most standard example of that is *intuitionistic* linear logic, which designates the same family of systems but in two-sided presentation, with always exactly one formula on the right. This effectively forbids the rules for $\parr$ and negation, so these systems include linear implication $\multimap$ as a primary connective, as in exercise 21. Such systems are named like the *classical* systems, except that "LL" is replaced by "ILL". Section 5.1 elaborates on MILL, the minimal intuitionistic system.

3 A bit of semantics

3.1 Provability semantics

The cut elimination property provides an argument for consistency of linear logic since it provide a computational interpretation of proofs and implies that the system is not degenerate. However, it provides no interpretation of formulas that would allow for the formulation of a completeness theorem meaning that there are enough rules for the connectives of the logic. In classical logic, this role is played by Boolean algebras, with the fact that a formula is provable in LK if and only if its interpretation is true in any such algebra. The proper notion in linear logic is that of phase spaces.

36 **Definition**. A *phase space* is a pair $(M, \bot)$ where M is a commutative monoid and $\bot$ is a subset of M. Two points $x, y \in M$ are called *orthogonal* if $xy \in \bot$. For a subset $A \subseteq M$, let define *the orthogonal* of A as $A^\bot := \left\{ y \in M \;\middle|\; \forall x \in A, xy \in \bot \right\}$. A *fact* is any subset of M of the form $A^\bot$.

Points of M are interpreted as tests, or possible interactions between a potential proof of a formula and a potential refutation of it. Elements of $\bot$ are *successful* tests or interactions in this set. Formulas will be interpreted by facts, which play the role of truth values and describe the set of possible behaviours for potential proofs.

This is parametricity in $\bot$ the key point in phase spaces; it is where the notion of interaction appears explicitly in the logic and how it differs from classical systems with traditional Boole-like semantics.

37 **Exercise**. Prove that, for any subsets A and B of M, $A \subseteq B$ implies $B^\bot \subseteq A^\bot$. Prove that for any A, $A \subseteq A^{\bot\bot}$ and $A^{\bot\bot\bot} = A^\bot$.

38 **Definition**. Let $(M, \bot)$ be a phase space and let e be the neutral element of M. For any subsets $A, B \subseteq M$, define

$$A \otimes B := \left\{ pq \;\middle|\; p \in A, q \in B \right\}^{\bot\bot} \qquad A \parr B := (A^\bot \otimes B^\bot)^\bot$$
$$A \oplus B := (A \cup B)^{\bot\bot} \qquad A \,\&\, B := A \cap B \qquad 0 := \varnothing^{\bot\bot} \qquad \top := M$$
$$!A := (A \cap I)^{\bot\bot} \qquad ?A := (A^\bot \cap I)^\bot \qquad 1 := \{e\}^{\bot\bot}$$

where I is the set of idempotent elements of M that are elements of the set 1.

An interpretation $[\![\cdot]\!]_M$ is defined by the choice of a fact $[\![\alpha]\!]_M$ for each propositional variable α. If A is a formula of LL, its interpretation $[\![A]\!]_M$ is deduced using the definitions above.

It is easy to check that if propositional variables are interpreted as facts, then for any formula A the interpretation $[\![A]\!]_M$ is a fact. Moreover, linear negation is indeed interpreted by the orthogonal. The interpretation of linear implication is instructive:

$$A \multimap B = A^\bot \parr B = \left\{ x \in M \;\middle|\; \forall y \in A, xy \in B \right\}$$

A point x is in $A \multimap B$ if, whenever it is composed with a point of A, the result is in B.

39 **Theorem**. *A formula A of LL is provable if and only if $e \in [\![A]\!]_M$ in all interpretations in phase spaces.*

Proof. Soundness is easily checked by induction on proofs. For completeness, the idea is to take for M the set of all sequents, considered up to permutation and up to duplication of ? formulas, with concatenation of sequents as the monoid operation. For $\bot$ we take the set the provable sequents. Then one checks by induction that for any formula A the set $[\![A]\!]_M$ is the set of all Γ such that $\vdash \Gamma, A$ is provable, if we take this as a definition for propositional variables. The neutral element e of M is the empty sequent, so $\vdash A$ is provable when $e \in [\![A]\!]_M$. □

Remark that, as a degenerate case, we can always choose $\bot = \varnothing$, then the set of facts is the elementary Boolean algebra $\{\varnothing, \top\}$. In this case, formulas are interpreted by their naive truth value, both conjunctions are interpreted as the classical one, similarly for disjunctions, all linearity is lost and we get back classical logic.

3.2 Proof semantics in coherence spaces

The idea of linearity historically appeared in the study of coherence spaces as a denotational model of the λ-calculus. Hence it is natural that proofs in linear logic have an interpretation in these spaces. We first define the interpretation of formulas as coherence spaces.

40 **Definition**. Let A and B be coherence spaces. The coherence spaces $A^\perp$, $A \otimes B$, $A \oplus B$ and $!A$ are defined as follows:

- $\left|A^\perp\right| = |A|$ and $x \frown_{A^\perp} x'$ if either $x = x'$ or x and x' are not coherent in A.
- $|A \otimes B| = |A| \times |B|$ and $(x, y) \frown_{A \otimes B} (x', y')$ if $x \frown_A x'$ and $y \frown_B y'$.
- $|A \oplus B| = (\{1\} \times |A|) \cup (\{2\} \times |B|)$ and $(i, x) \frown_{A \oplus B} (j, x')$ if $i = j$ and $x \frown x'$.
- $|!A|$ is the set of finite cliques of A and $x \frown_{!A} x'$ if $x \cup x'$ is a clique in A.

The spaces $A \parr B$, $A \,\&\, B$ and $?A$ are defined by duality.

The definition by duality can be easily reformulated directly for $A \parr B$ and $A \,\&\, B$. In particular, if we expand the definition for $A \multimap B$, we get that $(x, y) \frown_{A \multimap B} (x', y)$ if, whenever $x \frown_A x'$, we have $y \frown_B y'$: cliques in $A \multimap B$ are relations that map coherent points to coherent points. If the coherence spaces A and B are *flat* (i.e. coherence is equality) then cliques in $A \multimap B$ are graphs of partial functions from A to B.

A proof of a sequent $\vdash A_1, ..., A_n$ will be interpreted by a clique in the coherence space $A_1 \parr \ldots \parr A_n$. We will derive judgments of the form $\vdash \alpha_1 : A_1, ..., \alpha_n : A_n$ where each α_i is a point in A_i, meaning that the tuple $(\alpha_1, ..., \alpha_n)$ is an element of the interpretation of the considered proof. The coherent interpretation, in table 4 then appears as a decoration of the proof system.

Remark the particular shape of the rules for &: it appears as identical to the rules for $\oplus$, and indeed the difference comes form the fact that a proof $A \,\&\, B$ does contain a proof of A and a proof of B, and the interpretation makes a disjoint union of the interpretations of these proofs. Another point to remark is that, in the interpretation of the cut rule, it is required that both premisses use the same point α; the other rules have no such constraint.

41 **Theorem**. *The set of tuples in the interpretation of a proof is always a clique.*

Proof. By a simple induction of proofs. □

42 **Theorem**. *The interpretation of proofs in coherence spaces is invariant by cut elimination.*

Proof. By case analysis on the various cases of cut elimination. □

Identity

$$\frac{}{\vdash \alpha : A^\perp, \alpha : A}\ \text{ax} \qquad\qquad \frac{\vdash \gamma : \Gamma, \alpha : A \qquad \vdash \alpha : A^\perp, \delta : \Delta}{\vdash \gamma : \Gamma, \delta : \Delta}\ \text{cut}$$

Multiplicatives

$$\frac{\vdash \gamma : \Gamma, \alpha : A \qquad \vdash \beta : B, \delta : \Delta}{\vdash \gamma : \Gamma, (\alpha, \beta) : A \otimes B, \delta : \Delta}\ \otimes \qquad\qquad \frac{\vdash \gamma : \Gamma, \alpha : A, \beta : B}{\vdash \gamma : \Gamma, (\alpha, \beta) : A \parr B}\ \parr$$

Additives

$$\frac{\vdash \gamma : \Gamma, \alpha : A_i}{\vdash \gamma : \Gamma, (i, \alpha) : A_1 \,\&\, A_2}\ \&_i \qquad\qquad \frac{\vdash \gamma : \Gamma, \alpha : A_i}{\vdash \gamma : \Gamma, (i, \alpha) : A_1 \oplus A_2}\ \oplus_i$$

Exponentials

$$\frac{\vdash \gamma : \Gamma, \alpha : A}{\vdash \gamma : \Gamma, \{\alpha\} : ?A}\ ? \qquad \frac{\vdash \gamma : \Gamma}{\vdash \gamma : \Gamma, \varnothing : ?A}\ \text{w} \qquad \frac{\vdash \gamma : \Gamma, a : ?A, a' ?A}{\vdash \gamma : \Gamma, a \cup a' : ?A}\ \text{c}$$

$$\frac{\left\{\vdash a_{1,i} : ?A_1, \ldots a_{n,i} : ?A_n, b_i : B\right\}_{i \in I}}{\vdash \bigcup_{i\in I} a_{1,i} : ?A_1, \ldots \bigcup_{i\in I} a_{n:i} : ?A_n, \{b_i \mid i \in I\} : !B}\ !$$

Table 4: Interpretation of proofs in coherence spaces

4 A bit of proof theory

4.1 Intuitionistic and classical logics as fragments

In section 1.2, we introduced coherence spaces as a denotational semantics for the λ-calculus. The study of this model revealed the notion of linearity, from which linear logic was built, and proofs in linear logic also got an interpretation in coherence spaces in section 3.2. It is natural to ask how these models relate from a logical point of view. The answer to this question lies in the definition of the trace of a stable function, as of definition 10. It can be formalized as the definition of a translation from intuitionistic natural deduction into linear logic.

43 **Definition**. The translation $(\cdot)^*$ of formulas from implicative-conjunctive propositional logic into linear logic is defined inductively as

$$(\alpha)^* := \alpha \qquad (A \Rightarrow B)^* := !A^* \multimap B^* \qquad (A \wedge B)^* := A^* \,\&\, B^*$$

Sequents are translated as $(A_1, \ldots, A_n \vdash B)^* := \vdash ?(A_1^*)^\perp, \ldots, ?(A_n^*)^\perp, B^*$.

If we interpret this at the level of coherence spaces, we see that a point in $(A \Rightarrow B)^*$ is a point in $?(A^*)^\perp \parr B^*$, that is a pair of a clique in $(A^*)^\perp$ and a point in B^*. Developing further, we see that a clique in $(A \Rightarrow B)^*$ is a coherent set of such pairs, which is the condition for being the trace of a stable function.

This translation of formulas is naturally extended as a translation of proofs, which defines a translation of λ-terms into sequential proofs in linear logic. The introduction rule for $\Rightarrow$ exactly corresponds to the introduction rule for $\parr$, but the elimination rule requires more work:

$$\frac{\Gamma \vdash A \Rightarrow B \qquad \Delta \vdash A}{\Gamma, \Delta \vdash B} \quad \rightarrow \quad \dfrac{\dfrac{\vdash ?(\Gamma^*)^\perp, ?A^\perp \parr B \qquad \dfrac{\dfrac{\vdash ?(\Delta^*)^\perp, A}{\vdash ?(\Delta^*)^\perp, !A}\,! \qquad \dfrac{}{\vdash B^\perp, B}\,\text{ax}}{\vdash ?(\Delta^*)^\perp, !A \otimes B^\perp, B}\,\otimes}{\vdash ?(\Gamma^*)^\perp, ?(\Delta^*)^\perp, B}\,\text{cut}$$

44 **Exercise**. Write the translation of all typing rules of the λ-calculus into linear logic for this translation of formulas.

This translation of formulas has the crucial property that it preserves provability: a sequent $\Gamma \vdash A$ is provable in intuitionistic natural deduction if and only if its translation is provable in linear logic. For this reason, when dealing with sequents that are translations of intuitionistic ones, it is not a restriction, with respect to provability, to only accept proofs that are translations of intuitionistic proofs. Hence intuitionistic can be considered as a fragment of linear logic The simply typed λ-calculus can be seen as a fragment of MELL, since the purely implicative part does not use additives.

If we study cut elimination in this fragment, we subsequently get a decomposition of the β-reduction of λ-calculus into simpler steps. We will not develop this point in the present notes, however it does provide insights on the operational semantics of the λ-calculus and in particular makes the status of linearity explicit.

This translation of intuitionistic logic into linear logic suggests the possibility of also translating classical logic. It is well known that double-negation translations and their refinements map classical logic into intuitionistic logic; translation into linear logic simplifies the translation a bit further.

The crucial simplification is that, as for the translation $(A \Rightarrow B)^* = !A^* \multimap B^*$, everything happens through the use of modalities, carefully introduced in the translation in order to allow contraction and weakening when required by the original logic. In a classical sequent $\Gamma \vdash \Delta$, these rules are allowed both on the left (hence the ! on the left of linear implication) and on the right, which requires a ? on the right of linear implication. Translating $A \Rightarrow B$ into $!A \multimap ?B$ does not work, and systematic study reveals two possible translations:

$$(A \Rightarrow B)^t := !?A^t \multimap ?B^t \qquad \text{or} \qquad (A \Rightarrow B)^q := !A^q \multimap ?!B^q,$$

and variations around these (for instance, replacing ! with !! would not change the result very much). The translation extends to conjunction and disjunction by choosing the appropriate variant (additive or multiplicative) and requires some care; we will dot develop it here.

45 **Theorem**. *A sequent is provable in classical sequent calculus if and only if its translation in linear logic, by any of the above translations, is provable.*

Proof. Preservation of provability through the translation is proved by providing a translation of classical proof rules, in a similar way as for the intuitionistic case above. For the reverse implication, it suffices to remark that there is a simple reverse translation that erases modalities, translates all conjunctions to $\wedge$ and all disjunctions to $\vee$ and that this translation also preserves provability. □

This theorem implies that classical logic too can be considered as a fragment of linear logic. Actually, since there are two different translations here, two fragments of LL correspond to classical logic (for these translations, these systems are known as LKT and LKQ respectively). The study of these fragment applied to classical extensions of the λ-calculus reveal that the dichotomy between the two forms of translations is a proof-theoretical form of the computational dichotomy between call-by-name (for the t variant) and call-by-value (for the q variant). See Danos, Joinet, and Schellinx, "A new deconstructive logic: linear logic" for a thourough study of such translations.

4.2 Cut elimination and proof equivalence

The proof rewriting system for cut elimination is an operation of *normalization* for proofs. However, confluence of this system fails, but for bad reasons: depending on the order in which we apply cut elimination steps, we might end up with proofs that differ in the order of rules (moreover, the various uses of the exchange rules is unspecified, but this is a secondary problem). So we only get confluence up to commutation of independent rules.

On the other hand, any denotational semantics that is an invariant of cut elimination must interpret these equivalent proofs as equal. The point of proof nets, as introduced in section 5, is to avoid this problem by providing a formalism without rule commutation. We can also study what these commutations mean for sequential proofs.

We could indeed prove that the various rewriting rules for cut elimination form a confluent system modulo this quotient, so a given proof always has one cut-free form up to commutation, and it is obtained by the cut-elimination procedure (the details for proving this are not of particular interest in the context of these notes and we will stay at a more informal level). Lemma 24 then implies *strong normalization*, which means that any sequence of reductions eventually yields the same result. Hence the following definition is consistent.

46 **Definition**. For two proofs $\pi : \Gamma \vdash A, \Delta$ and $\pi' : \Gamma', A \vdash \Delta$, let $[\pi, \pi']$ be *the* cut-free form up to rule commutation of the proof obtained by applying the cut rule on π and π'.

This defines a notion of composition for cut-free proofs. We can elaborate on this notion by defining a proper category of cut-free proofs: the objects are the formulas of MLL and the morphisms from A to B are the cut-free proofs of $A \vdash B$. The composition operation of the above definition provides composition in the category, the axiom rule provides the identity morphisms. It is an easy but tedious lemma to prove the associativity of this composition.

Many models identify proofs a bit more than what cut elimination imposes. Consider these three proofs of the identity for $A \oplus (B \otimes C)$:

1. A simple axiom rule:

$$\dfrac{}{\vdash A^\perp \,\&\, (B^\perp \parr C^\perp), A \oplus (B \otimes C)}\ \text{ax}$$

2. An axiom rule for $B \otimes C$:

$$\dfrac{\dfrac{\dfrac{}{\vdash A^\perp, A}\ \text{ax}}{\vdash A^\perp, A \oplus (B \otimes C)}\ \oplus_1 \qquad \dfrac{\dfrac{}{\vdash B^\perp \parr C^\perp, B \otimes C}\ \text{ax}}{\vdash B^\perp \parr C^\perp, A \oplus (B \otimes C)}\ \oplus_2}{\vdash A^\perp \,\&\, (B^\perp \parr C^\perp), A \oplus (B \otimes C)}\ \&$$

3. Axiom rules for propositional variables only:

$$\dfrac{\dfrac{\dfrac{}{\vdash A^\perp, A}\ \text{ax}}{\vdash A^\perp, A \oplus (B \otimes C)}\ \oplus_1 \qquad \dfrac{\dfrac{\dfrac{\dfrac{}{\vdash B^\perp, B}\ \text{ax} \quad \dfrac{}{\vdash C^\perp, C}\ \text{ax}}{\vdash B^\perp, C^\perp, B \otimes C}\ \otimes}{\vdash B^\perp \parr C^\perp, B \otimes C}\ \parr}{\vdash B^\perp \parr C^\perp, A \oplus (B \otimes C)}\ \oplus_2}{\vdash A^\perp \,\&\, (B^\perp \parr C^\perp), A \oplus (B \otimes C)}\ \&$$

Along these lines, we could systematically define *expanded* proofs of the identity for any formula, the set of logical rules is such that this expansion is always possible.

- Associativity and commutativity

$$(A \oplus B) \oplus C \simeq A \oplus (B \oplus C) \qquad (A \otimes B) \otimes C \simeq A \otimes (B \otimes C)$$
$$A \oplus B \simeq B \oplus A \qquad A \oplus B \simeq B \oplus A$$
$$A \oplus 0 \simeq A \qquad A \otimes 1 \simeq A$$

- Distributivity

$$A \otimes (B \oplus C) \simeq (A \otimes B) \oplus (A \otimes C) \qquad A \otimes 0 \simeq 0$$

- Exponentiation

$$!(A \,\&\, B) \simeq !A \otimes !B \qquad !\top \simeq 1$$

- For any A and B, $A \simeq B$ iff $A^\perp \simeq B^\perp$.

Table 5: Standard isomorphisms in LL

47 This property is valid in any reasonable proof system, and if we look at it from the point of view of terms in the λ-calculus, we get the standard notion of η-expansion. For this reason, we keep the same terminology in the present context. We use the symbol $=_\eta$ to represent proof equivalence modulo rule commutation and η-expansion.

48 **Definition**. Two formulas A and B are *isomorphic* (written $A \simeq B$) if there are proofs π of $A \vdash B$ and ρ of $B \vdash A$ such that $[\pi, \rho]$ is η-equivalent to the axiom for $A \vdash A$ and $[\rho, \pi]$ is η-equivalent to the axiom for $B \vdash B$.

Among the linear equivalences that we have seen so far, many are actually isomorphisms. Table 5 enumerates the standard ones for LL. The additives and the exponentials provide examples of linear equivalences that are not isomorphisms:

$$A \oplus A \multimap A \qquad !A \otimes !A \multimap !A \qquad !!A \multimap !A \qquad !?!?A \multimap !?A$$

49 **Exercise**. Explain why these are not isomorphisms.

4.3 Reversibility and focalization

The proof rule for $⅋$ suggests that in a sequent $\vdash A_1, ..., A_n$, the comma means mostly the same thing as $⅋$. This is consistent with the definitions of the semantics of proofs, and it is also reminiscent of the fact that, in intuitionistic and classical logic, $A \vdash B$ means mostly the same thing as $A \Rightarrow B$ (and in linear logic it means $A \multimap B$, which is the same as $A^\perp ⅋ B$). This property has an interesting consequence about the structure of proofs.

50 **Proposition**. *Every proof of a sequent $\vdash \Gamma, A ⅋ B$ is equivalent modulo commutations of rules to a proof of $\vdash \Gamma, A, B$ followed by a $⅋$ rule.*

Proof. This can be established by induction on the initial proof of $\vdash \Gamma, A ⅋ B$ by showing that the $⅋$ rule that introduces $A ⅋ B$ can be commuted down with any rule coming after it. If $A ⅋ B$ is introduced by an axiom rule, this axiom can be η-expanded so that the $⅋$ is effectively introduced by a $⅋$ rule. □

This property is known as *reversibility*. It is remarkable that ⅋ is not the only reversible connective, indeed half of the connectives enjoy this property. The following theorem is a simple generalization of the above proposition and is no harder to prove.

51 **Theorem**. *The connectives ⅋, &, ⊥, and ⊤ are reversible.*

The dual connectives have an associated property called *focalization*, as defined below. For the purpose of explanation, it is useful to introduce the notion of *polarity*, which is based on these observations.[5]

52 **Definition**. A formula is *positive* if its main connective is ⊗, ⊕, 1, 0 or !. It is *negative* if its main connective is ⅋, &, ⊥, ⊤ or ?.

The reversibility property for negative formulas states in any proof, modulo rule commutations, one may assume that all negative connectives are introduced in the last rules.

53 **Theorem**. *Let $\Gamma = P_1, ..., P_n$ be a provable sequent consisting of positive formulas only. Then there is a formula P_i and proof of $\vdash \Gamma$ of the form*

$$\dfrac{\pi_1 \vdash \Gamma_1, N_1 \quad \dots \quad \pi_k \vdash \Gamma_k, N_k}{\vdash \Gamma_1, ..., \Gamma_k, P_i}\ R$$

where the N_j are the maximal negative subformulas of P_i and the last set of rules R builds P_i from the N_j.

In other words, in a positive sequent, there is a "principal" formula P_i such that we may assume that the last rules of the proof build P_i in one big step, without affecting formulas in the context. Hence, the proof has a "focus" on P_i.

5 Proof nets

The study of sequent calculus proofs and their dynamics in cut elimination is not a straightforward task, mostly because the system has heavy notations and a lot of technical details, like commutation rules, that do not feel crucial but make things obfuscated. In proof theory, natural deduction is often a preferred formalism because it is indeed more natural and easier to manipulate, as illustrated by the symptomatic example of the λ-calculus, which is a compact and efficient syntax for natural deduction in intuitionistic logic. This calculus contains the essence of intuitionistic logic and its cut elimination dynamics, as witnessed by Böhm's separation theorem and similar results. One would like a similar tool for linear logic, and proof nets are the answer to this question.

5.1 Intuitionistic LL and natural deduction

What makes the Curry-Howard correspondence work in λ-calculus is the intrinsic asymmetry of intuitionistic logic: from some set of hypothesis (the free variables) deduce one statement (the term, or rather its result). This asymmetry is a restriction with respect to classical logic but it allows nice simplifications and better property that stem from the functional nature of the resulting system.

The same restriction can also be applied in linear logic: consider the original two-sided presentation and impose that exactly one formula appears on the right of each sequent. The rules for negation and multiplicative disjunction are now forbidden, hence the associated connectives are dropped. However, linear implication $\multimap$, which is usually defined using them, is acceptable if we use the rules of exercise 21. The system we get is *multiplicative intuitionistic linear logic*, in short MILL.

[5]The focalization property was introduced in Andreoli, "Proposition pour une synthèse des paradigmes de la programmation logique et de la programmation par objets" with motivations related to computational interpretation of proof search in linear logic. It led to the definition of polarities in Girard, "A new constructive logic: classical logic" as properties of formulas in classical logic.

Identity

$$\frac{}{x : A \vdash x : A}\ \text{ax}$$

Implication

$$\frac{\Gamma, x : A \vdash t : B}{\Gamma \vdash \lambda x.t : A \multimap B}\ \multimap\text{R} \qquad \frac{\Gamma \vdash t : A \multimap B \quad \Delta \vdash u : A}{\Gamma, \Delta \vdash (t)u : B}\ \multimap\text{E}$$

Tensor

$$\frac{\Gamma \vdash t : A \quad \Delta \vdash u : B}{\Gamma, \Delta \vdash (t, u) : A \otimes B}\ \otimes\text{R} \qquad \frac{\Gamma, x : A, y : B \vdash t : C \quad \Delta \vdash u : A \otimes B}{\Gamma, \Delta \vdash t(x,y{:=}u) : C}\ \otimes\text{E}$$

Table 6: Intuitionistic linear logic as natural deduction

54 **Definition**. The language of formulas of MILL is defined by the following grammar:

$A, B :=$	α	propositional variable
	$A \multimap B$	linear implication — read "A implies B"
	$A \otimes B$	multiplicative conjunction — read "A tensor B"

Proof terms are defined by the following grammar:

$t, u :=$	x	variable — axiom
	$\lambda x.t$	linear abstraction — introduction of $\multimap$
	$(t)u$	linear application — elimination of $\multimap$
	(t, u)	pair — introduction of $\otimes$
	$t(x,y{:=}u)$	matching — elimination of $\otimes$

where x and y are taken from a set of variables. The variable x is bound in $\lambda x.t$, the variables x and y are distinct and bound in $t(x,y{:=}u)$. Standard conventions apply: bound names are assumed to be distinct from free names and terms are considered up to renaming of bound names.

A context (written Γ or Δ) is a partial mapping from variables to formulas with a finite domain. Proof terms are typed according to the rules of table 6, where Γ, Δ stands for the union of contexts, with the assumption that they have disjoint domains.

Remark that the fragment without the tensor and its associated constructs is exactly the λ-calculus, with contexts handled in multiplicative style, restricted so that contraction and weakening are forbidden. As a direct consequence, in typed terms, each variable that is either bound or declared in the context has exactly one occurrence in the term. The tensor rules provide a simple extension with pairs.

55 **Lemma**. *Let $t[u/x]$ denote the term t where the variable x is replaced by the term u. If the typings $\Gamma, x : A \vdash t : B$ and $\Delta \vdash u : A$ hold and Γ and Δ have disjoint domains, then $\Gamma, \Delta \vdash t[u/x] : B$ holds.*

Proof. Straightforward induction on the typing of t. □

In other words, the following cut rule is admissible:

$$\frac{\Gamma, x : A \vdash t : B \quad \Delta \vdash u : A}{\Gamma, \Delta \vdash t[u/x] : B}\ \text{cut}$$

This allows the definition of the cut elimination reduction using substitution, in a way similar to the β-reduction of usual λ-calculus.

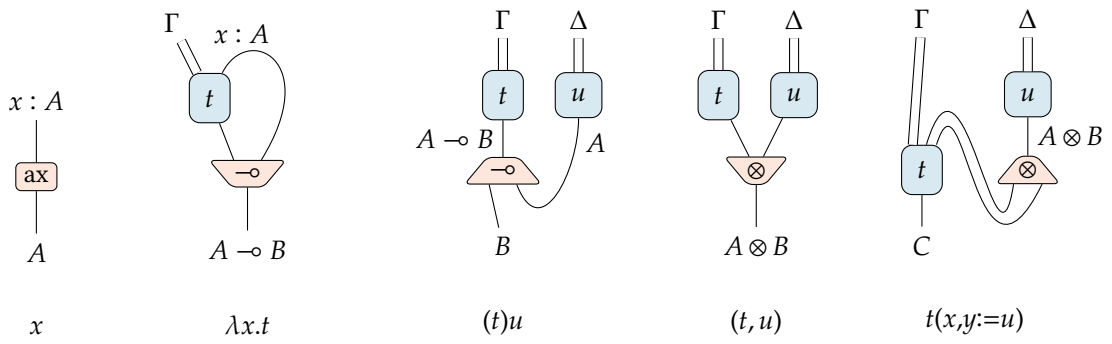


Figure 1: Graphical notation for MILL proof terms

56 **Definition**. Cut elimination for MILL is the congruent binary relation $\rightsquigarrow$ over proof terms generated by the following rules:

$$(\lambda x.t)u \rightsquigarrow t[u/x] \qquad\qquad t(x,y:=(u,v)) \rightsquigarrow t[u/x][v/y]$$

57 **Theorem**. *Cut elimination in MILL is strongly normalizing.*

Proof. Confluence of $\rightsquigarrow$ is easy to prove: contrary to the case of usual λ-calculus, this relation in MILL is strongly confluent (i.e. it has the diamond property) thanks to linearity.

Termination is also easy: each step strictly decreases the number of rules in the typing of the terms. Indeed, $(\lambda x.t)u \rightsquigarrow t[u/x]$ removes three rules ($\multimap$R, $\multimap$E, and the axiom for x) and $t(x,y:=(u,v)) \rightsquigarrow t[u/x][v/y]$ removes four rules ($\otimes$R, $\otimes$E, and the axioms for x and y). □

The structure of terms and cut-elimination may be illustrated using graphical notations. In figure 1, we propose such a notation: a term is represented by a graph, with nodes corresponding to logical rules and edges corresponding to occurrences of formulas in a proof. Orientation is relevant in our notations: dangling edges above a graph represent hypotheses, there is one such edge for each element of the typing context; the unique dangling edge below is the conclusion of the proof. Nodes pointing down represent introduction rules and nodes pointing up represent elimination rules, edges between nodes represent how each rule relates formulas, with the intuition that inputs are above and outputs are below.

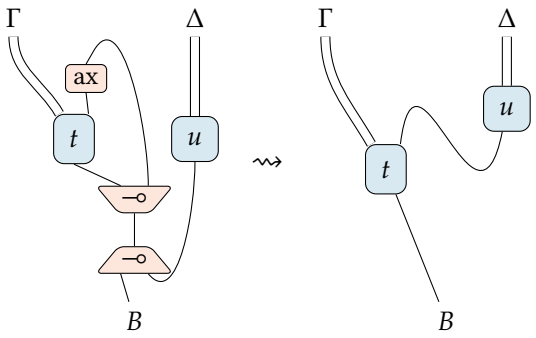
Cut elimination rules in this context are formulated as graph rewritings, as shown in figure 2. They occur when the output of an introduction rule is connected to the input of an elimination rule. Indeed, because of the intuitionistic structure of MILL, there is no need for commutation rules. Interaction steps appear as purely local graph rewriting operations: the two interacting rules plus the associated axiom rules are replaced by edges connecting the relevant parts of proofs. Note in particular that, again thanks to linearity, substitution is a very simple and local operation.

5.2 Proof structures

We now extend the graphical language of proofs to the full system of MLL. The transition from the intuitionistic fragment to the full one-sided system follows the same path as the transition between these systems in sequent calculus:

- Firstly, we allow several formulas on the right hand side of sequents. Subsequently, in the graphical language, there will be any number of outputs (or conclusions, i.e. dangling edges below). This allows for the reintroduction of $\parr$, which graphically simply joins together two conclusions of a proof, and linear negation, which transforms a hypothesis into a conclusion or vice-versa.

Linear implication:



Tensor:

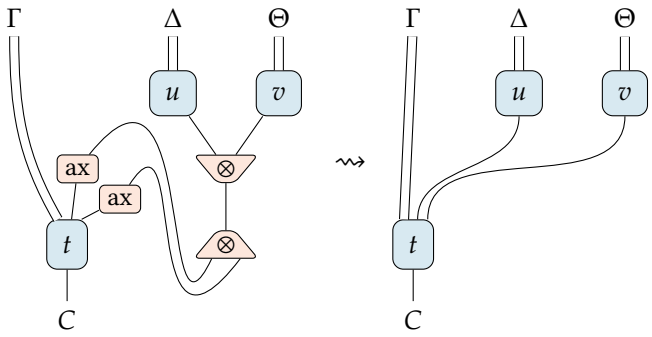


Figure 2: Cut elimination steps in MILL

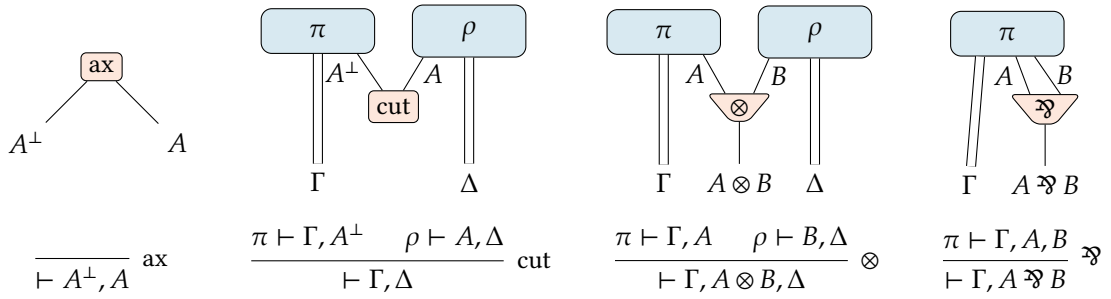


Figure 3: Translation of MLL sequent calculus rules into proof structures.

- Secondly, we hard-wire De Morgan duality, so that negation becomes an operation on formulas and sequents become one-sided. Graphically, this means that there are no inputs (i.e. dangling edges above) anymore and that axiom and cut now involve two edges bearing opposite formulas.

The four rules of MLL are translated graphically as shown in figure 3 (where $\pi \vdash \Gamma$ states the fact that π is a graph representing a proof of conclusion Γ). Formally, the kind of graphs built with these rules are called *proof structures*.

58 **Definition**. An MLL proof structure is a directed multigraph with edges labelled by MLL formulas and nodes labelled by either rule names or the special symbol "c". Moreover, for each node, incoming edges (called premisses) and outgoing edges (called conclusions) are equipped with a total order.

The node of each label imposes constraints on the number of incoming and outgoing edges as well as their labels, according to the corresponding proof rules. The labels must match the following constraints:

node label	c	ax	cut	$\otimes$	$\parr$
labels of incoming edges	A	$-$	$A, A^\perp$	A, B	A, B
labels of outgoing edges	$-$	$A^\perp, A$	$-$	$A \otimes B$	$A \parr B$

The nodes labeled "c" are called the *conclusions* of the structure.

In the translation of proof trees into proof structures given by figure 3, there are some implicit elements with respect to the above definition. Firstly, all edges are oriented from top to bottom. Secondly, conclusion nodes (labelled "c") are not drawn, instead we write the label of their incoming edge. We use double lines leading to a sequence Γ or Δ to represent an unspecified number of conclusions.

Finally, the natural handling of conclusion nodes when building structures is also implicit. For instance, in the rule for $\otimes$, the final structure is built using two structures π and ρ, introducing a new node labelled $\otimes$, changing the target of the edges for the premisses A and B so that they lead to this new node instead of their conclusion in π and ρ, introducing a new conclusion node for $A \otimes B$ and removing the conclusions nodes for A and B.

59 *Example.* A variant of the semi-distributivity rule from exercise 16, equivalent by commutativity of $\otimes$ and $\parr$, is written $(C \parr B) \otimes A \vdash (A \otimes B) \parr C$. In one-sided form, it is thus written $\vdash (C^\perp \otimes B^\perp) \parr A^\perp, (A \otimes B) \parr C$. A possible sequential proof of this sequent, with its translation as a proof structure, is given in figure 4. Note that we have placed the various nodes correctly so that the edges do not intersect, but this is just a matter of graph representation.

60 In this example, if we only look at the proof structure, we cannot deduce the sequential proof it was deduced from. Indeed, we get the same structure if we permute the two $\parr$ rules (this is consistent with the reversibility property described in section 4.3) and we also get the same structure if we exchange the order of the $\otimes$ rules. This defines four sequential proofs that have the same proof structure, and by

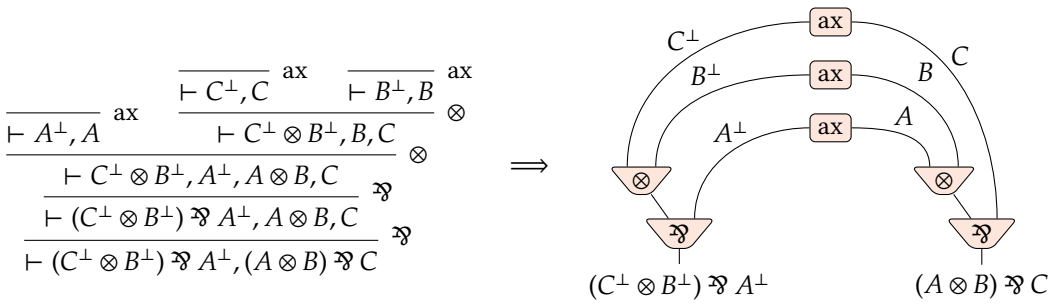


Figure 4: Example of a sequential proof and its translation as a proof structure.

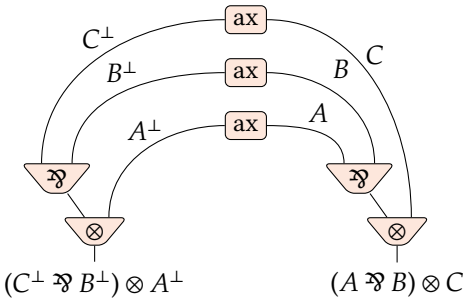


Figure 5: An incorrect proof structure.

case analysis it is easy to check that there can be no other sequential proof for this sequent and that all such proofs lead to this structure.

61 Approaching the question differently, one could try to enumerate all cut-free proof structures that have the conclusions $(C^\perp \otimes B^\perp) \parr A^\perp$ and $(A \otimes B) \parr C$ (assuming that A, B and C are propositional variables). Clearly the proof structure of figure 4 is the only one we can write.

The situation would be different if, say, C was replaced by B. In this case, there would be two different structures, depending on which premiss $B^\perp$ of the left tensor is connected to which premiss B on the right. These would correspond to two different classes of sequential proofs of $\vdash (B^\perp \otimes B^\perp) \parr A^\perp, (A \otimes B) \parr B$ that are not identified by the translation into structures.

62 **Definition**. A proof net is a proof structure that is the translation of some sequential proof.

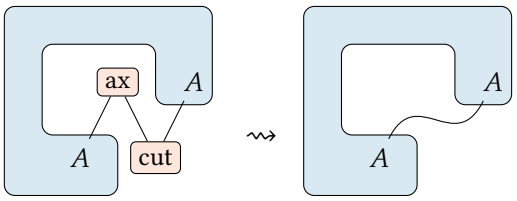
This notion refers to the fact that not any proof structure is actually the translation of a sequential proof, because the definition of proof structures only mentions the *local* structure of each logical rule. Indeed, the constraints for $\otimes$ and $\parr$ are exactly the same, while their sequent calculus counterparts are fundamentally different since $\parr$ has one premiss but $\otimes$ has two.

63 *Example.* The proof structure of figure 5, a variant of the one of figure 4 with $\otimes$ and $\parr$ exchanged, is not a proof net. To establish this fact, we can simply prove that the conclusion sequent $\vdash (C^\perp \parr B^\perp) \otimes A^\perp, (A \parr B) \otimes C$ is not provable without cut, since the structure of figure 4 has no cut. This is done by case analysis on the last rule of a hypothetical cut-free proof of this sequent:

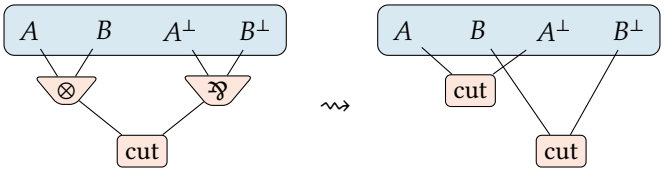
$$\frac{\vdash C^\perp \parr B^\perp, \Gamma \qquad \vdash A^\perp, \Delta}{\vdash (C^\perp \parr B^\perp) \otimes A^\perp, (A \parr B) \otimes C}\,\otimes \qquad \text{or} \qquad \frac{\vdash \Gamma, A \parr B \qquad \vdash \Delta, C}{\vdash (C^\perp \parr B^\perp) \otimes A^\perp, (A \parr B) \otimes C}\,\otimes$$

for some Γ and Δ. In the first rule, among Γ and Δ, one must be empty and the other must be $(A \parr B) \otimes C$. In either case, the premisses do not respect the linearity property for propositional variables

Axiom (assuming the right premiss of the cut node is not the left conclusion of the axiom node):



Tensor versus par:



Plus the same rules with the left and right premisses of the cut exchanged.

Figure 6: Cut elimination steps in MLL.

(proposition 17) so they cannot be provable. A similar argument applies to the other possibility for the last rule.

In this example, we already see that the crucial point for the tensor is that it *splits* the context in two, which implies splitting the proof in two independent subproofs.

64 **Exercise**. Enumerate all the cut-free proof structures with conclusions $(A^\perp \otimes A^\perp) \parr A^\perp$ and $(A \otimes A) \parr A$ and identify which ones are proof nets.

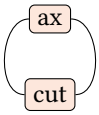
We now have a graphical language in which we can represent proofs, in a way that identifies sequential proofs that differ only by commutation of independent rules. In this setting, cut elimination steps are much simpler than in sequential proofs because only interaction rules are relevant: each cut has premisses that are introduction nodes or axiom nodes.

65 **Definition**. Cut elimination for MLL proof structures is the binary relation $\rightsquigarrow$ over proof structures such that $\pi \rightsquigarrow \rho$ if and only if ρ is obtained from π by applying one of the local graph rewriting rules of figure 6.

66 **Proposition (Strong normalization)**. *In any MLL proof structure, all maximal sequences of cut elimination steps are finite, all have the same length and they all reach the same irreducible proof structure (up to graph isomorphism).*

Proof. The finiteness of cut elimination sequences is obvious since each step strictly decreases the number of nodes. That all sequences have the same length and reach the same irreducible structure is easy since the relation is strongly confluent. The only critical pairs are when a cut is between the conclusions of two distinct axiom nodes and when an axiom has both its conclusions connected to different cuts, and in these cases the reducts are isomorphic. □

Observe that this property does not imply cut elimination, because there is one case of cut that is not reducible:



This kind of "vicious circle" is not a proof net, and thankfully so since it would be a proof of the empty sequent, since it has no conclusion node! Subsequently, it is clear that an irreducible proof net is always cut-free.

This leaves one question open in order to prove cut elimination for proof nets: how do we know that the reduct of a proof net is always a proof net, and not an incorrect proof structure? Correctness criteria provide the answer to this question.

5.3 Correctness criteria

A correctness criterion is a characterization of correct proofs among proof structures. In other words, it must characterize whether a given proof structure is *sequentializable*, i.e. if there is a sequential proof of which it is the graphical translation. Several such criteria exist in the literature, for various fragments of linear logic, motivated by various uses:

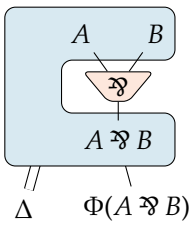
- It should be reasonably easy to prove that correctness is preserved by cut elimination.
- The efficiency of actually computing whether a structure satisfies the criterion should be known (and minimized), because it is directly related to the complexity of the decision problem for the considered logic.

Although the original criterion is the so-called *long trip* criterion, we will present here the Danos-Regnier criterion,[6] which is in the standard one in practice and is used as reference in the development of refined criteria.

As a gentle introduction, we will approach the problem of sequentializability from what we know of sequential proofs. Let us consider an arbitrary MLL proof structure π. The first interesting thing is reversibility of $\parr$, as shown in theorem 51. It implies that π is sequentializable if and only if it is the image of a sequential proof that ends with the introduction of any $\parr$ that is in conclusion. In other words, $\parr$ in conclusion are irrelevant for correctness.

We can actually generalize this property to any $\parr$ rule, not only in the conclusion of a proof. Consider a cut-free proof that contains a $\parr$ rule acting on two formulas A and B; since the proof has no cut, $A \parr B$ is a subformula of a conclusion, which we can write $\Phi(A \parr B)$:

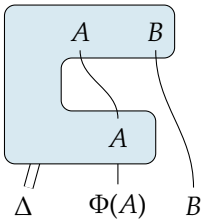
$$\dfrac{\dfrac{\dfrac{\vdots}{\vdash \Gamma, A, B}}{\vdash \Gamma, A \parr B}\,\parr}{\vdots}{\vdash \Delta, \Phi(A \parr B)}$$



It is easy to check that the $\parr$ rule can be removed, which changes the conclusions but not the validity of the proof. Instead of $A \parr B$, we have two formulas A and B, and we may choose to apply the same rules, not to $A \parr B$ but to A, for instance, leaving B as an extra conclusion:

[6] This criterion was introduced in Danos and Regnier, "The structure of multiplicatives" together with other criteria, as part of a thorough study of proofs in MLL.

$$\dfrac{\dfrac{\dfrac{\vdots}{\vdash \Gamma, A, B}}{\vdots}}{\vdash \Delta, \Phi(A), B}$$



Of course, we could have made the opposite choice, kept B inside the proof and turned A into an extra conclusion. Such a choice for a $\parr$ is called a *switching*.

67 **Definition**. A switching s of a proof structure π is a function mapping each $\parr$-labelled node of π to either 1 or 2. The switched structure $s(\pi)$ is is the structure obtained by replacing each $\parr$ node n in π by an edge from its $s(n)$-th premiss to its conclusion and its other premiss to a new conclusion node.

As a consequence of the above remark, the operation of switching a $\parr$ in a proof structure preserves correctness. If we switch all $\parr$ rules (assume there are n of them), then we get new proof structures with no cut and no $\parr$ (there are 2^n of them, if we consider all possible switchings).

68 **Lemma**. *If π is a correct cut-free proof structure, then for all switchings s of π, $s(\pi)$ is correct.*

Now, what can we say about a sequentializable proof structure that only contains axiom and $\otimes$ nodes? The shape of the translation of a $\otimes$ rule (as in figure 3) shows that applying such a rule means taking two proof nets and connecting one node of each to a new node (which is connected to a fresh conclusion). Because there are two independent proof nets as premisses in the rule, this cannot create a cycle (even in the underlying undirected graph). Moreover, if the premisses were connected graphs, then the resulting structure is connected too. This is the property we are after.

69 **Lemma**. *A proof structure with only axiom and $\otimes$ nodes is correct if and only if, as an undirected graph, it is acyclic and connected.*

Proof. The direct implication is justified by the above remarks and can be proved by induction on a sequential proof. The reverse implication is proved by recurrence on the number of $\otimes$ nodes in the structure, it suffices to remark that removing any $\otimes$ node connected to a conclusion effectively disconnects the structure into two connected components, which are correct by recurrence hypothesis; the base case of no $\otimes$ node corresponds to a single axiom node, which is obviously a correct proof. □

This condition of acyclicity and connectedness of switchings is the Danos-Regnier criterion.

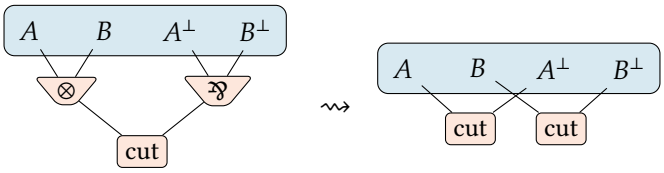
70 **Theorem (Danos-Regnier)**. *An MLL proof structure is sequentializable if and only if all its switchings are acyclic and connected.*

Sketch of proof. The "only if" part is essentially contained in the arguments above. For the "if" part, the key point is to prove that the condition does imply the existence of a splitting $\otimes$ node in a structure with only $\otimes$ as conclusions that respects the criterion. There are several methods to establish this in the literature but we will not elaborate on this point in the present notes. □

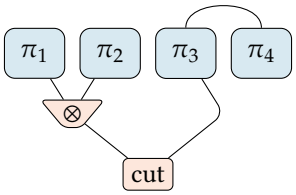
Note that the above theorem does not exclude the presence of cut nodes. Indeed, cuts do not change the problem significantly: by similar arguments as above, one can justify that a structure with cuts is correct if and only if the same structure where cuts are replaced by tensor rules is correct. This is a consequence of the fact that the tensor and cut rule have strictly the same geometrical structure (two subproofs as premisses, with one formula taken from each).

71 **Theorem**. *Cut elimination preserves correctness of proof structures.*

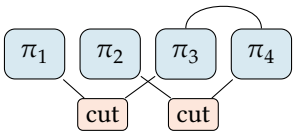
Proof. The case of the elimination of the axiom is straightforward, so we focus on the case of tensor versus par. We have a reduction with the shape



such that all switchings of the left-hand side are acyclic and connected, and we must show that the switchings of the right-hand side are acyclic and connected too. Consider a switching s of the right-hand side. It can be extended as s_1, the switching of the left-hand side that keeps the first premiss, and s_2, the other way around. The switched structure for s_1 must have the shape



where the π_i are connected and acyclic subgraphs, with π_4 connected to one of the other components because of connectedness of $s_1(\pi)$. If π_4 was connected to π_1 or π_2, then in the switching s_2 we would have a cycle, since the tensor is connected to π_4, but that cannot happen by hypothesis. Hence π_4 is necessarily connected to π_3 and in the reduct we have



which is acyclic and connected. Hence the reduct satisfies the Danos-Regnier criterion, so it is a proof net. □

This fills the missing part in the proof of cut elimination for proof nets, since we had strong normalization in theorem 66. As a consequence, proof nets are a proper syntax for proofs in MLL for which we can prove the key properties directly, without reference to proof trees. The graphical formulation gets rid of all unimportant commutations between rules, so we can even consider that proof nets are *canonical* proof objects.

The extension of this technique to wider fragments of linear logic is a rich topic, with various open questions. The formulation of proof nets as canonical as in MLL is often unknown but approximations exist.

- The introduction of multiplicative units requires some technique to formulate the proper notion of proof net, because naive extensions with a $\perp$ node break the connectivity condition.
- For MELL, the situation is almost as good, except that the promotion rule forces sequentiality in the form of *boxes*: each instance of this rule, together with its subproof, becomes a single node in the proof net, with as many outputs as there are formulas in the conclusion of the rule; the node is associated with an independent proof net for the subproof (the "contents" of the box).

- Additive connectives are known to be difficult, and the proper solution has been found nearly twenty years after the introduction of proof nets, in Hughes and Glabbeek, "Proof nets for unit-free multiplicative-additive linear logic".

References

Andreoli, Jean-Marc. "Proposition pour une synthèse des paradigmes de la programmation logique et de la programmation par objets". Thèse de doctorat. Université de Paris VI, 1990.

Danos, Vincent, Jean-Baptiste Joinet, and Harold Schellinx. "A new deconstructive logic: linear logic". In: *Journal of Symbolic Logic* 62 (1996), pp. 755–807.

Danos, Vincent and Laurent Regnier. "The structure of multiplicatives". In: *Archive for Mathematical Logic* 28 (1989), pp. 181–203.

Girard, Jean-Yves. "A new constructive logic: classical logic". In: *Mathematical Structures in Computer Science* 1.3 (1991), pp. 255–296.

— "Linear Logic". In: *Theoretical Computer Science* 50 (1987), pp. 1–102.

— "Linear Logic: Its Syntax and Semantics". In: *Advances in Linear Logic*. Ed. by Jean-Yves Girard, Yves Lafont, and Laurent Regnier. Cambridge University Press, 1995, pp. 1–42.

— "The system F of variable types, fifteen years later". In: *Theoretical Computer Science* 45.2 (1986), pp. 159–192. DOI: http://dx.doi.org/10.1016/0304-3975(86)90044-7.

Girard, Jean-Yves, Yves Lafont, and Paul Taylor. *Proofs and types*. Cambridge University Press, 1989.

Hughes, Dominic J. D. and Robert J. van Glabbeek. "Proof nets for unit-free multiplicative-additive linear logic". In: *18th IEEE Symposium on Logic in Computer Science (LICS)*. June 2003, pp. 1–10.

Lincoln, Patrick et al. "Decision problems for propositional linear logic". In: *Annals of Pure and Applied Logic* 56.1-3 (1992), pp. 239–311. DOI: http://dx.doi.org/10.1016/0168-0072(92)90075-B.